DOCUMENT RESUME

ED 175 685 SE 028 673

AUTHOR Scheid, Francis J., Ed.
TITLE Supplementary and Enrichment Series: 1+1=?. SP-27.
INSTITUTION Stanford Univ., Calif. School Mathematics Study Group.
SPONS AGENCY National Science Foundation, Washington, D.C.
PUB DATE 66
NOTE 103p.: For related documents, see SE 028 648-675; Contains occasional light and broken type

EDRS PRICE MF01/PC05 Plus Postage.
DESCRIPTORS Curriculum; *Enrichment; *Games; *Instruction; *Mathematical Applications; Mathematics Education; *Puzzles; Secondary Education; *Secondary School Mathematics; Supplementary Reading Materials
IDENTIFIERS *School Mathematics Study Group

ABSTRACT

This is one in a series of SMSG supplementary and enrichment pamphlets for high school students. This series is designed to make material for the study of topics of special interest to students readily accessible in classroom quantity. Topics covered include the basic rules of sequence arithmetic, elementary strategy, intermediate strategy, and advanced strategy. (MP)

SCHOOL MATHEMATICS STUDY GROUP

SP-27

SUPPLEMENTARY and ENRICHMENT SERIES

1 + 1 = ?

Edited by Francis J. Scheid

PHOTOLITHOPRINTED BY CUSHING - MALLOY, INC.
ANN ARBOR, MICHIGAN, UNITED STATES OF AMERICA

Financial support for the School Mathematics Study Group has been provided by the National Science Foundation.

PREFACE

Mathematics is such a vast and rapidly expanding field of study that there are inevitably many important and fascinating aspects of the subject which, though within the grasp of secondary school students, do not find a place in the curriculum simply because of a lack of time.

Many classes and individual students, however, may find time to pursue mathematical topics of special interest to them. This series of pamphlets, whose production is sponsored by the School Mathematics Study Group, is designed to make material for such study readily accessible in classroom quantity.

Some of the pamphlets deal with material found in the regular curriculum but in a more extensive or intensive manner or from a novel point of view. Others deal with topics not usually found at all in the standard curriculum. It is hoped that these pamphlets will find use in classrooms in at least two ways. Some of the pamphlets produced could be used to extend the work done by a class with a regular textbook but others could be used profitably when teachers want to experiment with a treatment of a topic different from the treatment in the regular text of the class. In all cases, the pamphlets are designed to promote the enjoyment of studying mathematics.

Prepared under the supervision of the Panel on Supplementary Publications of the School Mathematics Study Group:

1+1=?

A question which leads us back to an old, but increasingly attractive, description of both abstract and applied mathematics, and also ahead to the modern problems of computing machine design. In two parts.

PART ONE: ABSTRACTION

Chapter 1

A STATEMENT OF OBJECTIVES

You can't blame parents for being a little confused nowadays about what their children are studying in the name of mathematics. One day it seems that

$$1 + 1 = 2$$

which is comforting and traditional, but on another day one hears that

$$1 + 1 = 10$$

which is simple enough when you've been let in on the secret. But then there's 'throw away two' arithmetic, in which

$$1 + 1 - 0$$

and, as you'll soon discover, a strong case can also be made for

$$1 + 1 = 1.$$

No wonder teachers are often asked what 'modern mathematics' is all about.

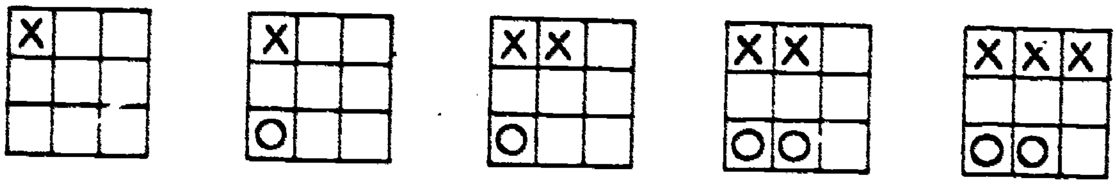
All four of the above answers to the question '1 + 1 = ?' will eventually find a place in these chapters. But before even starting on the details let's take just a few moments for a description of long-range objectives. There are many ways of trying to explain what mathematics is really 'all about.' One very brief definition, having a certain element of surprise, has over the centuries received the blessing of several big-name philosophers. It reads simply,

'Mathematics is a collection of games.'

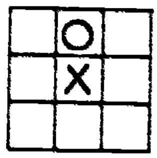
Now those six words shouldn't be expected to carry too heavy a message, and hopefully some further explanation will be welcomed. It could run like this.

> 'In order to play any game you first have to learn certain basic rules. The basic rules are the things you need to know even to understand what particular game is being played. For the familiar, and fairly dull, game of Tic-Tac-Toe, three basic rules are more or less sufficient.
>
> 1. Players take turns.
> 2. Put your mark in an empty space.
> 3. First three in a row wins.
>
> You need to know these basic rules if you're going to play Tic-Tac-Toe at all. But if this is all that you know, you could play the game very

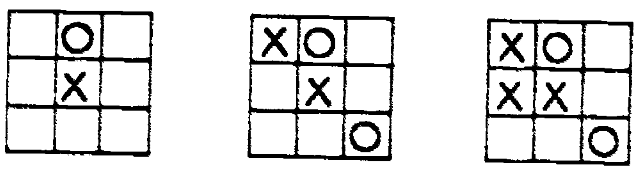
badly. Here is an example of legal, but uninspired, play.



It's a win for X, but the competition was pitiful. Nobody plays Tic-Tac-Toe this badly for long. As in any other game, once you know the basic rules you try to figure out the <u>strategy</u> of the game, the things that are nice to know in order to play skilfully. Here is an example of strategy in Tic-Tac-Toe. Player X has taken the center space and player O a side.



X can now guarantee himself a win, in quite a variety of ways. The logic is easy, and all Tic-Tac-Toe players soon think their way through victories such as this.



X is now prepared to win either in column one or in row two, and O will not be able to block both. Knowing this strategy brings X his win.

This sort of progress, from basic rules to strategy, is typical of most games. It's true of chess, poker, tennis and almost any game you can think of. It's also true of the various parts of mathematics. Arithmetic and geometry, which everyone agrees are parts of mathematics, involve both basic rules and strategy. It's the custom in mathematics to call the basic rules <u>axioms</u> and the strategy <u>theorems</u>. But the pattern is the same. Starting from the basic rules (or axioms) you try to figure out the higher strategy (or theorems) of the game, so that you'll be able to play skilfully. The confusion in 'modern mathematics' is due to the fact that there are lots of games other than arithmetic and geometry, which are also parts of mathematics, but seldom reached the

public eye until recently. Children are now becoming familiar with some of these games, but to parents who never had the chance they must look strange.'

Hopefully, this longer explanation of what mathematics is will be helpful. But to reinforce it, to get a clear view of the game-like nature of mathematics, a detailed study of one of its parts is essential. In the following chapters one of the most important, but lesser known, games of modern mathematics is presented. It is called 'sequence arithmetic.' Its basic rules are explained first, as with any game, followed by examples of amateur play. Then some strategy is developed and play becomes more skilfull. In spite of its importance, sequence arithmetic is a much easier game than the more familiar 'ordinary arithmetic,' and just where its place in the modern curriculum will eventually prove to be, whether in kindergarten or college, makes interesting speculation.

Two other points should be mentioned before we start. First, the games that constitute mathematics are not played just for fun. There is definitely some good entertainment value in them, as mathematicians will testify, but there are also ways in which the games contribute to the progress of human affairs. Exhibiting the possible areas of application of sequence arithmetic is a major objective in what follows. It is also the most interesting part of the story. And the second point is this. In mathematics you play strictly by the rules. Play is honest. Strategy is carefully checked to be sure that it is correct. This policy of honest play is a strong tradition in mathematics, inspired in part by the fact that so often the results are used in important applications. Demonstrating this policy of honest play is another major objective.

In summary, the objectives are:

1. to exhibit the game-like nature of mathematics;
2. to show that 'math is honest;'
3. to show that 'math is useful.'

It's all been said before, but goals are important enough to rate frequent repetition.

Chapter 2

THE BASIC RULES

2.1 The Rules Themselves.

The game of sequence arithmetic is played with sequences of zeros and ones. Two sample sequences are these, which have been named X and Y for short.

X	0 1 0 0 1 1
Y	1 1 0 1 0 1

The zeros and ones are called 'values,' so X and Y are both six values long. Don't try to attach any 'meaning' to such sequences, not yet anyway. Just think of a sequence as a thing to play with, like a tennis ball or a chess queen. There are three basic rules to learn before play can begin. The first one tells how to 'add' two sequences.

> BASIC RULE I: To add two sequences, put one over the other and deal with each pair of corresponding values separately, making
>
> $$\begin{array}{cccc} 0 & 0 & 1 & 1 \\ \underline{+0} & \underline{+1} & \underline{+0} & \underline{+1} \\ 0 & 1 & 1 & 1 \end{array}$$

Making 1 + 1 = 1 is the only surprise, and it shows that, whatever 1 stands for here, it isn't your old friend the <u>number</u> 1. As a first example of addition, here is the computation of the sum X + Y.

X	0 1 0 0 1 1
Y	1 1 0 1 0 1
X + Y	1 1 0 1 1 1

Addition is certainly a simple enough job. Notice that the sum X + Y is another sequence of zeros and ones, six values long, just as X and Y are. Another thing to notice is that it doesn't make any difference which sequence is put on top and which at the bottom. We just had X on top of Y, and called the sum X + Y. If we put Y on top of X

Y	1 1 0 1 0 1
X	0 1 0 0 1 1

the sum should probably be called Y + X instead. But computation produces

Y + X	1 1 0 1 1 1

which is absolutely identical with the X + Y sequence just computed. Basically this is because 0 + 1 and 1 + 0 have both been prescribed as 1. Plainly there's no need to fuss. Either sequence can be put on top, and we'll use X + Y and Y + X interchangeably.

Let's turn to the second basic rule, which introduces multiplication of sequences.

BASIC RULE II. To multiply two sequences put one over the other and deal with each pair of corresponding values separately, making

$$\begin{array}{cccc} 0 & 0 & 1 & 1 \\ \underline{\times 0} & \underline{\times 1} & \underline{\times 0} & \underline{\times 1} \\ 0 & 0 & 0 & 1 \end{array}$$

This time there are no surprises at all. As a first example of multiplication here is the computation of the product XY.

X	0 1 0 0 1 1
Y	1 1 0 1 0 1
XY	0 1 0 0 0 1

Multiplying sequences is as easy a job as adding them. Notice that the product XY is another sequence of zeros and ones, six values long, just as X and Y are. Notice also that once again it makes no difference which sequence is put on top and which at the bottom. We just had X on top of Y, and called the product XY. If we put Y on top of X

Y	1 1 0 1 0 1
X	0 1 0 0 1 1

the product should probably be called YX instead. But computation produces

YX	0 1 0 0 0 1

which is absolutely identical with the XY sequence just computed. Basically this is because 0 × 1 and 1 × 0 have both been prescribed as 0. Plainly there's no need to fuss. Either sequence can be put on top and we'll use XY and YX interchangeably.

These first two basic rules are simple enough, but if you find it easier to remember words than symbols, here is a brief translation. 'The only way to get 0 in a sum is from 0 + 0, and the only way to get 1 in a product is from 1 × 1.' We're up to the third and last basic rule.

> BASIC RULE III. To invert a sequence, replace 1 by 0 and replace 0 by 1.

This hardly needs an example, but here is the computation of the inverse of X, represented by the symbol $\overline{X}$, which you read 'X inverse.'

X	0 1 0 0 1 1
$\overline{X}$	1 0 1 1 0 0

Notice that X inverse is another sequence of zeros and ones, six values long, just as X is.

2.2 First Calisthenics.

Now you know all the basic rules of sequence arithmetic, and it's time for some first efforts at playing the game. Start by computing these three sequences, using the X and Y sequences of the previous section.

$$\overline{Y}$$

$$\overline{X}\ \overline{Y}$$

$$\overline{X + Y}$$

The long bar over X + Y indicates the inverse of the X + Y sequence which was computed earlier. As a check on your arithmetic, the last two sequences ought to be the same. In symbols,

$$\overline{X + Y} = \overline{X}\ \overline{Y}.$$

The equality symbol as used here means 'is the same sequence as.' So $\overline{X + Y}$ ought to have turned out to be the same sequence as $\overline{X}\ \overline{Y}$. If you disagree, then check the basic rules and examples once more to see if you've misunderstood, because both of these sequences should have come out

0 0 1 0 0 0.

A question often asked by beginners in sequence arithmetic is whether $\overline{X}\ \overline{Y}$ and $\overline{X\ Y}$ are the same. ($\overline{X\ Y}$ means the inverse of the product X Y.) To find out in the case of our particular X and Y sequences, compute the inverse of XY

$$\overline{X\ Y}$$

and compare it with $\bar{X}\,\bar{Y}$. They should be different. But then compute

$$\bar{X} + \bar{Y}$$

and compare it with $\overline{X\,Y}$. They should be the same. In symbols

$$\overline{X\,Y} = \bar{X} + \bar{Y}.$$

Now compute the sums

$$X + X$$
$$Y + Y$$
$$\bar{X} + \bar{X}$$

and then try the products.

$$X\,X$$
$$Y\,Y$$
$$\bar{X}\,\bar{X}$$

When you're finished you'll agree that sequence arithmetic has some very simple features. Another simple feature is illustrated by the addition of a few sequences to their own inverses,

$$X + \bar{X}$$
$$Y + \bar{Y}$$
$$X\,Y + \overline{X\,Y}$$

and still another by these multiplications.

$$X\,\bar{X}$$
$$Y\,\bar{Y}$$

Making these various computations may begin to suggest some strategy, or theory of the game of sequence arithmetic, but we'll postpone theory until the next chapter. Before leaving our X and Y sequences, here is a final set of arithmetical calisthenics, intended mostly as a limbering-up exercise.

$$X + X\,Y$$
$$X\,Y + X\,\bar{Y}$$
$$\bar{X}\,Y + \bar{X}\,\bar{Y}$$

These three sequences should turn out identical. Then try three more.

$$X\,\bar{Y} + \bar{X}\,Y$$
$$(X + Y)(\bar{X} + \bar{Y})$$
$$X\,Y + \bar{X}\,\bar{Y}$$

These should also agree amongst themselves. If you disagree, double-check your computations.

2.3 The Shortest Sequences.

As further calisthenics let's switch to some very short sequences, first taking the shortest of them all. Our X and Y were six values long, but any length is suitable, the rules remaining the same. There are only two sequences of length one. One of them is

$$0$$

and the other is

$$1$$

and it almost seems like flattery to call them sequences at all. It's a simple job to prepare addition and multiplication tables for these miniature sequences. Here is the addition table.

+	0	1
0	0	1
1	1	1

It tells you that $0 + 1$ is 1, and that $1 + 1$ is 1, and so on. This is all very familiar to you by now. Complete the following multiplication table yourself.

×	0	1
0		
1		

It's also reasonably clear that 0 and 1 are inverses of each other, and that about exhausts the computational possibilities of these sequences.

Graduating just slightly, there are only four sequences of length two. Call them $\emptyset$, P, Q and I. (Read $\emptyset$ as OH.)

$\emptyset$	0	0
P	0	1
Q	1	0
I	1	1

With these little sequences computations can be done in your head. To get $P + Q$, for example, we can just look at P and Q, and apply Basic Rule I.

P + Q	1	1

You can see that $P + Q$ has turned out to be the sequence I.

$$P + Q = I$$

And what is P Q? Looking at the P and Q sequences again, and applying Basic Rule II, leads quickly to

P Q	0 0

so that P Q has turned out to be the sequence ∅.

$$P\,Q = \emptyset$$

These two results are recorded in the addition and multiplication tables below.

+	∅	P	Q	I
∅				
P		P	I	
Q	Q	I		
I				

×	∅	P	Q	I
∅		∅		
P			∅	
Q		∅		Q
I				

A few other entries have also been made. These entries claim that

$$Q + P = I \qquad Q\,I = Q$$
$$P + P = P \qquad \emptyset\,P = \emptyset$$
$$Q + \emptyset = Q \qquad Q\,P = \emptyset$$

and you should check to see if you agree. Then supply all the missing entries, completing the tables. Each place should be filled with ∅, P, Q or I.

There is also an inversion table for these little sequences,

	∅	P	Q	I
-		Q		

The entry which has been made claims that the inverse of P is Q. Do you agree? Supply the missing entries, writing either ∅, P, Q or I in each empty place. As you complete these three tables you will be forming some impressions of the strategy, or theory, of the game of sequence arithmetic.

For a final limbering up take the sequences of length three. There are exactly eight of them, and let's name them as follows.

∅	0 0 0	D	0 1 1
A	0 0 1	E	1 0 1
B	0 1 0	F	1 1 0
C	1 0 0	I	1 1 1

The addition and multiplication tables for these sequences are somewhat larger, of course. A few entries are included below, but most of the labor is left to you.

+	∅	A	B	C	D	E	F	I
∅	∅	A	B	C				
A	A	A	D	E				
B	B	D	B	F				
C	C	E	F	C				
D								
E								
F								
I								

×	∅	A	B	C	D	E	F	I
∅	∅	∅	∅	∅				
A	∅	A	∅	∅				
B	∅	∅	B	∅				
C	∅	∅	∅	C				
D								
E								
F								
I								

The inversion table for these sequences is simpler. Complete it.

	∅	A	B	C	D	E	F	I
-	I	F						

At this point you can consider yourself a skilful amateur at the game of sequence arithmetic.

Chapter 3

ELEMENTARY STRATEGY

3.1 Only Two Distinct Kinds of Column.

From your first attempts to play this game perhaps you're drawing some conclusions. You've used sequences of length one, two, three and six. Any length can be used, from one upward, the basic rules remaining the same. But in any one version of sequence arithmetic all sequences should have the same length. In other words, don't mix different lengths.

Now we come to the matter of strategy, or, as it's called in mathematics, theory. We'll develop a short list of theorems, just a part of a much longer list which is recorded in the literature of mathematics. You'll learn how to prove these theorems, and later we'll put them to work in applications. The theorems are true for sequences of any length, and we'll use different lengths at different times for examples. Whatever the length, two sequences are very important. One of them contains all zeros, no ones, and we'll call it $\emptyset$. (Read it OH.)

$\emptyset$: all 0's

The other sequence contains all ones, no zeros, and we'll call it I.

I: all 1's

The practice calculations of Chapter 2 may have suggested to you what can be expected whenever a sequence is multiplied by $\emptyset$, but here's one further example to make the picture totally clear. Pick some sequence and call it A. I'll choose this one.

A: 0 1 1 0 1 0

To multiply A by $\emptyset$ we put $\emptyset$ under A, as usual

$\emptyset$: 0 0 0 0 0 0.

The product comes out

A$\emptyset$: 0 0 0 0 0 0

and it's identical with $\emptyset$. Surely everybody begins to believe that the product will come out $\emptyset$ no matter what sequence A we start with. In other words, we suspect a theorem, a first piece of strategy for the game of sequence arithmetic.

THEOREM 1. $A\emptyset = \emptyset$.

In this brief statement of what we suspect to be true, the letter A stands for 'any sequence at all.' Translated into English prose the theorem becomes, 'When any sequence at all is multiplied by $\emptyset$, the product is $\emptyset$.' How can we be sure this theorem is true? Actually, a proof is very easy. Look at the example again. The thing to notice is that the first two columns alone tell the whole story. The other four columns are merely duplicates. If we chose an entirely different sequence for A, we'd get exactly the same two distinct kinds of column, the first two that we have in our example.

A	0 1 . . . .
$\emptyset$	0 0 . . . .
$A\emptyset$	0 0 . . . .

In these two columns the product $A\emptyset$ comes out zero. So, it will always come out zero, in every column, because all other columns are duplicates. This definitely proves that $A\emptyset$ comes out a solid sequence of zeros, no ones. In symbols, $A\emptyset = \emptyset$.

It has taken quite a few words, and some patience, to prove Theorem 1, which may have seemed perfectly obvious from the start. The reason for this patience is partly that, in mathematics at least, it pays to be very careful. But there's a more immediate reason, too. The easiest way to prove lots of the theorems of sequence arithmetic is by this idea of duplicating columns. The spirit of our proofs, for a while, will be, 'How many distinct kinds of column do we have to examine?' For Theorem 1, only two distinct kinds will appear, and that's true of this entire chapter.

Now we'll turn to a companion theorem, which is also discernable in the opening examples. Theorem 1 concerns multiplying a sequence by $\emptyset$. What happens when a sequence is added to I? Take any sequence at all, and call it A. Let's choose just the first two values of A and leave the rest open for a moment.

A: 0 1 . . . .

Add I to this sequence. I is just

I: 1 1 1 1 1 1 1 1 1

so the sequence comes out (remember, $1 + 1 = 1$)

A + I: 1 1 . . . .

and it's identical with I, at least in the first two columns. But almost at once you realize that it won't make any difference how the A sequence is completed. Either 0 or 1 will go into each open position and the new columns of computation will just be duplicates of the two columns we already have. The sum A + I will be a solid sequence of ones, no zeros. It will be the sequence I. This is our second theorem.

$\boxed{\text{THEOREM 2.}\quad A + I = I.}$

In this theorem, as in Theorem 1, A can be any sequence at all.

Since we're doing so well with the $\emptyset$ and I sequences, here is another pair of companion theorems which involve these two special sequences.

$\boxed{\text{THEOREM 3.}\quad A + \emptyset = A.}$

$\boxed{\text{THEOREM 4.}\quad AI = A.}$

The proofs of these two are left to you. Just follow the pattern by which Theorems 1 and 2 were proved. As in the earlier pair, A can be any sequence at all. Even so, only two distinct kinds of column can appea: .

3.2 <u>Theorem Guessing</u>.

Now let's break away from $\emptyset$ and I for a moment to pick up another pair of simple companion theorems. You may have guessed these two also in your practice session. Take any sequence at all, and call it A, perhaps the same A we used a moment ago.

A: 0 1 1 0 1 0

To calculate A + A, you put A under A,

A: 0 1 1 0 1 0

and eventually get the sum,

A + A: 0 1 1 0 1 0

which may be revealing enough for you to guess how to finish our next theorem.

$\boxed{\text{THEOREM 5.}\quad A + A = \quad .}$

If you can't guess, then the completed theorem appears at the end of this chapter, along with other results you will be asked to guess shortly. For example, calculating A times A should suggest the companion to Theorem 5. Can you guess it?

$\boxed{\text{THEOREM 6. } AA = \qquad .}$

If not, see the completed theorem at the end of this chapter. As usual, this pair is true for any A sequence at all, and once you've guessed the missing righthand sides, the proofs are easy. (No matter what sequence A is, only two distinct columns appear in the computation of A + A or AA.)

Next, let's figure out some of the strategy, or theory, of playing with inverses of sequences. The idea of inverting a sequence is simple enough; you just swap zeros for ones, and ones for zeros. So, there ought to be some simple theorems involving inverses. To start us off, complete this one.

$\boxed{\text{THEOREM 7. } \overline{\emptyset} = \qquad , \text{ and } \overline{I} = \qquad .}$

Unlike our first six theorems, in which the sequence A can be 'any sequence at all,' this just points out the inverses of our two special sequences $\emptyset$ and I. To return to the earlier spirit, take any sequence at all, and call it A. We might choose the first two values, leaving the rest of the sequence open for later choosing.

$$A:\ 0\ 1\ .\ .\ .\ .\ .$$

Invert the sequence,

$$\overline{A}:\ 1\ 0\ .\ .\ .\ .\ .$$

and then <u>invert the inverse</u>.

$$\overline{\overline{A}}:\ 0\ 1\ .\ .\ .\ .\ .$$

Notice the two bars over the A. Each bar says 'invert what's under me,' so $\overline{A}$ means the inverse of A, while $\overline{\overline{A}}$ means the inverse of $\overline{A}$. Now you can finish choosing the A sequence, and complete the calculation. Whatever you choose, it's easy to guess the theorem, true for any A sequence at all,

$\boxed{\text{THEOREM 8. } \overline{\overline{A}} = \qquad .}$

Before graduating to bigger things, here is a final pair of companion theorems, true for any A sequence at all, and provable by the method of 'only two distinct columns.' Complete them if you can.

THEOREM 9. $A + \overline{A} =$.

THEOREM 10. $A\overline{A} =$.

Simple as these first ten theorems are, they have an important role to play in further developing the strategy of the game of sequence arithmetic, and in making applications. To close up this chapter here is a summary of our 'elementary strategy,' including the theorems which were left unfinished back along the way. Remember, A can be 'any sequence at all.'

Theorem 1. $A\emptyset = \emptyset$.
Theorem 2. $A + I = I$.
Theorem 3. $A + \emptyset = A$.
Theorem 4. $AI = A$.
Theorem 5. $A + A = A$.
Theorem 6. $AA = A$.
Theorem 7. $\overline{\emptyset} = I$ and $\overline{I} = \emptyset$.
Theorem 8. $\overline{\overline{A}} = A$.
Theorem 9. $A + \overline{A} = I$.
Theorem 10. $A\overline{A} = \emptyset$.

PROBLEM. Using the following (incomplete) A sequence

A 0 1

and the special sequences

$\emptyset$ 0 0 0 0 0 0
I 1 1 1 1 1 1

compute the first two values of each of the following sequences.

$A\emptyset$	$\overline{\emptyset}$
$A + I$	$\overline{I}$
$A + \emptyset$	$\overline{A}$
AI	$\overline{\overline{A}}$
$A + A$	$A + \overline{A}$
AA	$A\overline{A}$

You now have the only two distinct kinds of column which can appear no matter how this A sequence is completed, or even if an entirely different A sequence is chosen. Sequences which are identical in these two columns will be identical for 'any A sequence at all.' Comparing $A\emptyset$ and $\emptyset$, in these two columns, we find they agree. Comparing $A + I$ with I, in these two columns, we find they agree. This is how Theorems 1 and 2 were proved. Now make similar comparisons (of $A + \emptyset$ with A, and of AI with A, and so on), thereby proving Theorems 3 to 10.

Chapter 4

INTERMEDIATE STRATEGY

4.1 Only Four Distinct Kinds of Column.

Now that you've seen some theorems which are true for any sequence A at all, a fairly natural next step is to look for theorems which are true for any pair of sequences. These are not usually so easy to guess, so we'll follow the way which our mathematical ancestors have cleared for us. To begin, you could choose any pair of sequences you wish. Call them A and B. An excellent choice is this pair, for reasons which will soon appear and which you may guess. These are only four values long.

A	0 0 1 1
B	0 1 0 1

First calculate their sum and product

$$A + B$$
$$AB$$

and their inverses.

$$\bar{A}$$
$$\bar{B}$$

Then calculate these four sequences.

$$\overline{A + B}$$
$$\bar{A}\,\bar{B}$$
$$\overline{AB}$$
$$\bar{A} + \bar{B}$$

(Inverse bars are made just long enough to stretch over the sequence to be inverted. Thus, $\overline{A + B}$ means the inverse of the sum $A + B$, while $\overline{AB}$ means the inverse of the product AB. You will see even longer bars in later chapters.) Now, if you've completed the above computations successfully, then $\overline{A + B}$ and $\bar{A}\,\bar{B}$ should be the same. They should both be

1 0 0 0

so that, at least for our special pair of A and B sequences, $\overline{A + B}$, (which is known as the inverse of the sum) and $\bar{A}\,\bar{B}$ (the product of the inverses) have turned out to be the same sequence. Next, compare $\overline{AB}$ with $\bar{A} + \bar{B}$. They should both be

so that $\overline{AB}$ (known as the inverse of the product) and $\bar{A} + \bar{B}$ (the sum of the inverses) have also turned out to be the same sequence. Let's also take a moment to notice two things that are *not* true. Many an amateur at sequence arithmetic has assumed that $\overline{A + B} = \bar{A} + \bar{B}$, and that $\overline{AB} = \bar{A}\,\bar{B}$. Notice that in your computations above both of these are false. Those computations seem to be forecasting this pair of theorems.

THEOREM 11. $\overline{A + B} = \bar{A}\,\bar{B}$.
THEOREM 12. $\overline{AB} = \bar{A} + \bar{B}$.

At least, these theorems are true for my choices of A and B.

Now we come to the important point. We chose sequences of length four, so naturally there are four columns of computation in the above work. If you choose longer A and B sequences, then you'll surely have more columns of computation. But, no matter which two sequences you choose for A and B, even if they are a hundred values long, computing $\overline{A + B}$, $\bar{A}\,\bar{B}$, $\overline{AB}$ and $\bar{A} + \bar{B}$ as I've suggested here *will not produce any new kind of column*. You'll get *more* columns, but they'll all be *duplicates* of these four. The four columns here are *the only distinct kinds* there are. That's because, whatever sequences you choose for A and B, the two top entries of each column must start the column off in one of these four ways

0 0 1 1
0 1 0 1

and the top two entries of a column determined what happens all the way down that column. Think it over, but only *four distinct kinds of column* are possible, regardless of how A and B are chosen, and the four kinds are exactly what we already have. We can deduce that Theorems 11 and 12 are true, not only for our special choices of A and B, but for *any A and B sequences at all*.

This method of only four distinct columns, though simple, is powerful. We'll use it to provide honest proofs of a number of theorems. First let's take a pair of companion theorems that were mentioned informally way back when basic rules I and II were first introduced. When two sequences are being added it's entirely immaterial which sequence goes above the other. The sum will be the same either way. To be absolutely precise it was suggested that we might distinguish between A + B and B + A, saying that when A is put above B then the sum is called A + B, and when B is put above A, then the sum is

called B + A. That would be clear and neat. But the fact that such fuss has no computational significance is the content of Theorem 13.

THEOREM 13. A + B = B + A.

The same remarks apply to products, and you could guess the companion of Theorem 13.

THEOREM 14. AB = BA.

As usual, A and B stand for any pair of sequences. These theorems are so easy to believe that it seems almost silly to write out proofs. Nevertheless, just to be safe, here is how the proofs look, using the same A and B sequences as before.

A	0 0 1 1	B	0 1 0 1
B	0 1 0 1	A	0 0 1 1
A + B	0 1 1 1	B + A	0 1 1 1
AB	0 0 0 1	BA	0 0 0 1

Comparing A + B with B + A, and AB with BA, we surely find that they agree. For these particular A and B sequences Theorems 13 and 14 are secure. But to repeat, other choices of A and B might lead to more columns of computation, but they can't possibly lead to new kinds of columns. The four columns we have here are the only four distinct kinds there are. Our proof for this A and B pair covers all other A and B pairs also.

The content of these two theorems must seem terribly 'obvious.' Unless you've grown accustomed to the spirit of modern mathematics, you may even object to dignifying them as theorems. But the point is, that theorems are strategy, and to play the game of sequence arithmetic honestly, it's important to list strategy which we know to be correct. Then, if we play according to our list, we know we're safe. The policy in mathematics is, 'Better safe than sorry,' and mathematicians have been led to this policy by hard experience. All too often, playing by intuition instead of by proven strategy, they've run into logical disaster. Intuition is great stuff, and should be worked to its limit, but whenever possible it should be checked by honest logic. Here we'll carefully list all proven strategy, and then we'll play by our list.

4.2 The Four Basic Products.

Now we come to a small avalanche of theorems. All of them involve four famous and basic products; namely, AB, $A\bar{B}$, $\bar{A}B$ and $\bar{A}\,\bar{B}$. The proofs of these theorems will be much easier if we examine these products first. Take the usual A and B sequences,

$$\begin{array}{ll} A & 0\,0\,1\,1 \\ B & 0\,1\,0\,1\,. \end{array}$$

Calculate the inverses,

$$\begin{array}{l} \bar{A} \\ \bar{B} \end{array}$$

and then the four famous products.

$$\begin{array}{c} AB \\ A\bar{B} \\ \bar{A}B \\ \bar{A}\,\bar{B} \end{array}$$

Notice the resulting pattern. Each product takes the value 1 in only one of our four columns. But between them the four products manage to supply a 1 for each column. The pattern can be used to simplify the job of proving theorems such as this.

$$\boxed{\text{THEOREM 15.}\quad AB + A\bar{B} = A.}$$

Everything needed for the proof is already available. Adding the products AB and $A\bar{B}$ should product the sequence

0 0 1 1

which definitely is the same as A. And to repeat once more, should you use other A and B sequences than mine, then $AB + A\bar{B}$ should still agree with A, because the computations above produce the only four distinct kinds of column that A and B sequences can generate. In these four columns $AB + A\bar{B}$ agrees with A, so it always will, whatever sequences you choose for A and B. That's the method of only four distinct columns. Apply this simple, but powerful, method to the following avalanche of theory, all of which is true for any pair of sequences A and B. Try proving these in your head, just looking at the four basic products up above, but witho further penmanship. If that's too much headwork, then put the computations do on paper, but be sure to prove them all, one way or another.

THEOREM 16. $A + AB = A$.

THEOREM 17. $A + B = A + \bar{A}B$.

THEOREM 18. $A + B = (AB + A\bar{B}) + \bar{A}B$.

THEOREM 19. $\bar{A}B + \bar{A}\,\bar{B} = \bar{A}$.

THEOREM 20. $(AB + A\bar{B}) + (\bar{A}B + \bar{A}\,\bar{B}) = I$.

THEOREM 21. $(A + B)(\bar{A} + \bar{B}) = A\bar{B} + \bar{A}B$.

THEOREM 22. $(A + B)\overline{AB} = A\bar{B} + \bar{A}B$.

In several of these theorems parentheses are used to designate which computations have priority. Do computations inside the parentheses first.

4.3 Only Eight Distinct Kinds of Column.

Now let's stretch the method of distinct columns once more, by proving a few theorems that are true for any three sequences A, B and C. Take these three special sequences first.

A	0 0 0 0 1 1 1 1
B	0 0 1 1 0 0 1 1
C	0 1 0 1 0 1 0 1

There's a fairly obvious pattern to these selections. And that pattern is useful, because it provides one neat way to guarantee that the eight columns of zeros and ones which are already beginning to shape up will be the only eight distinct kinds that are possible from three sequences A, B and C. You can choose longer sequences for A, B and C, and get more columns, but any new column will duplicate one of the eight we are starting to develop here. You should convince yourself of this before pushing onward. Try attaching more values to A, B and C. Any new column you form will be a duplicate of one of these eight. This fact is important because it means that whatever we prove for these special A, B, C sequences will hold for any three sequences at all. This is the method of only eight distinct kinds of column. Let's put it right to work. First compute

$$B + C$$

and then multiply by A.

$$A(B + C)$$

Next get the two products

$$AB$$

$$AC$$

and finally compute their sum.

$$AB + AC$$

Unless you've made an error in arithmetic you should find that $A(B + C)$ and $AB + AC$ have come out identical. Both should be

0 0 0 0 0 1 1 1

and that suggests a theorem. Since both of these sequences are the same for our special choice of A, B and C, they will also be the same for any other choice.

THEOREM 23. $A(B + C) = AB + AC.$

As usual, there is a companion theorem. Computing these five sequences

$$BC$$

$$A + BC$$

$$A + B$$

$$A + C$$

$$(A + B)(A + C)$$

you should again discover a pair of identical sequences. The companion theorem reads

THEOREM 24. $A + BC = (A + B)(A + C).$

and it's true for any three sequences A, B and C.

To close up this chapter let's apply the method of only eight distinct columns to a pair of theorems which are crucial for our work in the chapters ahead. The first one concerns 'double sums.'

THEOREM 25. $A + (B + C) = (A + B) + C.$

This is sometimes called a 'shift parentheses' theorem, and it's easy to see why. The parentheses designate which sequences are to be added first. On the left, $B + C$ should be computed first, and then A should be added to the sum. On the right, $A + B$ should be computed first, and then C should be added to that sum. The theorem guarantees that both orders of procedure lead to the

same final result. I'm sure that you're more than willing to believe this theorem, but in the interest of safety take a few moments to apply our method of only eight distinct columns.

A	0 0 0 0 1 1 1 1
B	0 0 1 1 0 0 1 1
C	0 1 0 1 0 1 0 1
B + C	
A + (B + C)	
A + B	
(A + B) + C	

The fifth and seventh sequences should be identical, and Theorem 25 is proved. The companion of Theorem 25 is another 'shift parentheses' theorem, but with products in place of sums.

THEOREM 26. A(BC) = (AB)C.

And needless to say, the proof is easy by the method of <u>only eight distinct kinds of column</u>. If you need further practice at writing zeros and ones, then compute BC, A(BC), and so on. Theorem 26 will stand up to the test. Further opportunities to apply this method will be provided in the next chapter.

Chapter 5

ADVANCED STRATEGY

5.1 New Strategy from Old.

After the fairly heavy dose of theory in the previous two chapters it would probably be good psychology to offer you a few applications. Certainly you have a right to know why this game of sequence arithmetic was invented, and what it's currently good for. If you're desperate to see the applications, turn at once to Chapter 6. But if you can stand a final (extra heavy) dose, this chapter will finish our penetration into the theory of sequence arithmetic. Then we'll give applications our undivided attention. And if our recent efforts cause spectres of sixteen or more distinct kinds of column to float across your imaginations, in a dense fog of zeros and ones, let me quickly reassure you. Such methods are feasible, but we're going to turn to a different technique of proof for our last theorems. We will use the strategy we already have to develop more advanced strategy. This method of getting new strategy from old is by far the best way for proving some of the theorems of this chapter. For other theorems the old method of distinct columns may still be the easier, but we'll use the new method anyway just for the experience.

As a first example, we know by Theorem 13 that

$$A + B = B + A$$

so that two sequences may be added in either order. Suppose we have three sequences to add, call them A, B and C. Then by Theorem 25

$$A + (B + C) = (A + B) + C.$$

This 'shift parentheses' theorem says that we can add B to C first, or we can add A to B first. The final sums will be the same. Could we even add A to C first? I'm sure you guess yes, and maybe you see how to prove it. Applying Theorem 13 to the sequences $(A + B)$ and C, we can lengthen our last result to

$$A + (B + C) = (A + B) + C = C + (A + B)$$

and now shifting parentheses in the rightmost member

$$A + (B + C) = (A + B) + C = (C + A) + B.$$

This shows that in adding A, B and C it makes no difference which pair of sequences we decide to add together first, B and C, or A and B, or

C and A. Moreover, having done this first addition, what remains is to add the two sequences that are left, such as A and (B + C). But we know that any two sequences can be added in either order, and so we deduce that any three sequences A, B and C may be added in any order we care to take them. The sum of three such sequences is usually written without any parentheses

$$A + B + C$$

to show that the order of addition is entirely immaterial. Let's call such a sum a 'double sum.'

If we turn next to triple sums

$$A + B + C + D$$

and to even longer sums, it's a natural guess that the order of computation is still immaterial. Hopefully, the following theorem is true.

THEOREM 27. Double, triple, and longer sums such as

$$A + B + C$$

$$A + B + C + D$$

and so on, may be computed in any order.

The proof of this theorem presents a new type of difficulty. The trouble is that sums can come in an endless variety of lengths, and we have to handle all lengths. For situations of this sort a one-step-at-a-time procedure proves to be useful. To illustrate, suppose that we knew the theorem to be true for sums involving five sequences. As a next step we could consider sums of six sequences, such as this one.

$$A + B + C + D + E + F$$

One way of tackling this sum is to add the A and B sequences first.

$$(A + B) + C + D + E + F$$

This leaves us with only five sequences, and so from here on the order of computation won't matter. But now suppose we begin again by adding D and F together first instead of A and B. Is it possible that the final result will be different? To find out we return to the five-sequence sum we already have. Since the order of computation in that sum doesn't matter, let's choose to take sequence D right after A and B.

$$(A + B) + D + \text{etc.}$$

The shift parentheses theorem instantly converts this to

$$A + (B + D) + \text{etc.}$$

which is a different, but equal, five-sequence sum, in which $B + D$ is computed first. Now let's change our minds. After computing $B + D$ let's take sequence F next, with A being postponed until later.

$$(B + D) + F + \text{etc.}$$

Again shift parentheses and you have

$$B + (D + F) + \text{etc.}$$

This sum is still the equal of the others we've had, and now $D + F$ is the first sum computed. So the question of a moment ago is answered. Adding D and F first leads to exactly the same final result we would get by adding A and B first. And the same sort of proof applies no matter which pair you want to add together first! The resulting sum will equal our original

$$(A + B) + C + D + E + F.$$

You may want to choose a few pairs and explore the details yourself.

Now we come to the main point. We've just proved that <u>if</u> five-sequence sums can be computed in any order <u>then</u> six-sequence sums can also be computed in any order. But these lengths serve only as an example. The idea of our proof works just as well for longer or shorter sums! What has actually been proved is that grafting an extra sequence to a sum doesn't upset the applecart. If the order of addition was immaterial before, then it remains immaterial after the graft. This is the crux of the one-step-at-a-time method. For instance, we do already know that sums of <u>three</u> sequences (double sums) can be computed in any order. The same is therefore true when there are <u>four</u> sequences involved. And if four-sequence sums can be computed in any order, then the same will be true of five-sequence sums. And so on it goes, up to sums of any length. You will want to think over the details of this proof of Theorem 27 fairly carefully, and to fill in details which were omitted for brevity. But basically it's an honest proof, and it certainly makes heavy use of earlier strategy. Needless to say, there is a companion theorem.

> THEOREM 28. Double, triple and longer products such as
>
> ABC
> ABCD
>
> and so on, may be computed in any order.

The companion is proved in the same way. Although these two theorems are not exciting, they are important. You'll be seeing many multiple sums and products from here on.

5.2 Some Special Products.

We already know that for any three sequences A, B and C

$$A(B + C) = AB + AC.$$

That's our Theorem 23. The A sequence multiplies B + C on the left, and it multiplies both B and C on the right. It isn't hard to guess how this theorem can be stretched for longer sums. First comes

$$A(B + C + D) = AB + AC + AD$$

which is true for any four sequences A, B, C and D. Here's a quick proof. There are two double sums involved, but we don't have to worry about the order in which we compute them, so start by giving B + C a temporary alias, X. Then

$$\begin{aligned} A(B + C + D) &= A(X + D) \\ &= AX + AD \\ &= A(B + C) + AD \\ &= AB + AC + AD \end{aligned}$$

The proof is already finished, and you'll have to admit it was speedier than tackling sixteen distinct kinds of columns would have been. We've simply used Theorem 23 twice. It's another example of using strategy already proved to develop more advanced strategy. The same idea can be applied again to stretch the sum another notch, and then another, and so on. It's the one-step-at-a-time method again in action. The indicated result is our next theorem.

THEOREM 29. For any sequences A, B, C, D, E, etc.

$$A(B + C + D) = AB + AC + AD$$
$$A(B + C + D + E) = AB + AC + AD + AE$$

and so on.

Since products can be taken in either order, all the products in Theorems 23 and 29 can be reversed.

$$(B + C)A = BA + CA$$
$$(B + C + D)A = BA + CA + DA$$
$$(B + C + D + E)A = BA + CA + DA + EA$$

So whether the sums on the left precede A or follow it, the A sequence appears in each part of the right hand sum. We can now stretch Theorem 29 in a second direction. As a first example, consider $(A + B)(C + D)$ and give $C + D$ a temporary alias, X.

$$\begin{aligned}(A + B)(C + D) &= (A + B)X \\ &= AX + BX \\ &= A(C + D) + B(C + D) \\ &= AC + AD + BC + BD\end{aligned}$$

The result is a triple sum. As one further example,

$$\begin{aligned}(A + B + C)(D + E + F) &= (A + B + C)X \\ &= AX + BX + CX \\ &= A(D + E + F) + B(D + E + F) \\ &\qquad + C(D + E + F) \\ &= AD + AE + AF + BD + BE + BF + CD \\ &\qquad + CE + CF\end{aligned}$$

and the sum on the right includes nine sequences. Here is the indicated theorem.

THEOREM 30. For any sequences A, B, C, D, E, etc.

$$(A + B)(C + D) = AC + AD + BC + BD$$

$$\begin{aligned}(A + B + C)(D + E + F) &= AD + AE + AF + BD + BE \\ &\qquad + BF + CD + CE + CF\end{aligned}$$

and so on.

PROBLEM 1. Prove $A(B + C + D + E) = AB + AC + AD + AE$ by giving $B + C$ the temporary alias X and then using Part 1 of Theorem 29, which we have already proved.

PROBLEM 2. Simplify Theorem 30 (Part 2) for the special case when $F = \emptyset$.

PROBLEM 3. Simplify Theorem 30 (Part 1) for the special case when $C = \overline{A}$ and $D = \overline{B}$. (Notice that this amounts to a second way of proving Theorem 21.)

PROBLEM 4. Simplify Theorem 30 (Part 2) for the special case when $D = \overline{A}$, $E = \overline{B}$ and $F = \overline{C}$.

5.3 Th̲e̲ E̲i̲g̲h̲t̲ B̲a̲s̲i̲c̲ P̲r̲o̲d̲u̲c̲t̲s̲.

Now that we have the crucial, though unexciting, Theorems 27 to 30 behind us we can move more quickly to accumulate the remaining items of advanced strategy that will be needed.

THEOREM 31. $A(B + C) = ABC + AB\bar{C} + A\bar{B}C$.

The proof of this by the method of eight distinct columns is simple enough, but for variety, here is a proof which uses our earlier accumulated strategy. Check each line carefully to be sure you agree.

$$\begin{aligned} A(B + C) &= AB + AC \\ &= ABI + AIC \\ &= AB(C + \bar{C}) + A(B + \bar{B})C \\ &= ABC + AB\bar{C} + ABC + A\bar{B}C \\ &= ABC + AB\bar{C} + A\bar{B}C. \end{aligned}$$

Theorems 4, 5, 9, 23, 27 and 28 all see action. Can you find where? Next comes a pair of theorems involving the double sum $AB + AC + BC$.

THEOREM 32. $AB + AC + BC = ABC + AB\bar{C} + A\bar{B}C + \bar{A}BC$.
THEOREM 33. $AB + AC + BC = AB + (A + B)\overline{AB}C$.

The proof of each of these will remind you of the proof just completed. First,

$$\begin{aligned} AB + AC + BC &= ABI + AIC + IBC \\ &= AB(C + \bar{C}) + A(B + \bar{B})C + (A + \bar{A})BC \\ &= ABC + AB\bar{C} + ABC + A\bar{B}C + ABC + \bar{A}BC \\ &= ABC + AB\bar{C} + A\bar{B}C + \bar{A}BC. \end{aligned}$$

And second, remembering Theorem 22 which says $(A + B)\overline{AB} = A\bar{B} + \bar{A}B$,

$$\begin{aligned} AB + (A + B)\overline{AB}C &= ABI + (A\bar{B} + \bar{A}B)C \\ &= AB(C + \bar{C}) + (A\bar{B} + \bar{A}B)C \\ &= ABC + AB\bar{C} + A\bar{B}C + \bar{A}BC \\ &= AB + AC + BC. \end{aligned}$$

The thing to notice is that, in all three proofs just given, certain basic products,

$$ABC \qquad AB\bar{C} \qquad A\bar{B}C \qquad \bar{A}BC$$

play key roles. Introducing the I sequence as was done, and then replacing I by $A + \bar{A}$ or $B + \bar{B}$ or $C + \bar{C}$, whichever seemed best, led us to combina-

tions of these basic products. In proving other results concerning three sequences A, B and C there are four other basic products which can appear.

$$A\,\overline{B}\,\overline{C} \qquad \overline{A}\,B\,\overline{C} \qquad \overline{A}\,\overline{B}\,C \qquad \overline{A}\,\overline{B}\,\overline{C}$$

Some of these appear in the proofs of Theorems 34 to 36, which you are asked to attempt as problems.

THEOREM 34. $A\overline{B} + B\overline{C} + C\overline{A} = \overline{A}B + \overline{B}C + \overline{C}A.$

THEOREM 35. $A\overline{B} + B\overline{C} + C\overline{A} = A\,\overline{B}\,C + A\,\overline{B}\,\overline{C} + A\,B\,\overline{C} + \overline{A}\,\overline{B}\,C + \overline{A}\,B\,C + \overline{A}\,B\,\overline{C}.$

THEOREM 36. The sum of all eight basic products is I.

To the relatively innocent bystander the two sides of Theorem 34 must look like inverses of each other. But they are equal in spite of appearances. Both equal the sum of basic products shown in Theorem 35. Theorem 36 is one indicator of the special role which the eight basic products play. It is a close relative of Theorems 9 and 20.

$$A + \overline{A} = I$$

$$AB + A\overline{B} + \overline{A}B + \overline{A}\,\overline{B} = I.$$

PROBLEM 5. Prove Theorem 34 by showing that each side is the same combination of six basic products. This also proves Theorem 35.

PROBLEM 6. Prove Theorem 20 in a second way, starting with

$$I = II = (A + \overline{A})(B + \overline{B})$$

and then using Theorem 30.

PROBLEM 7. Continuing Problem 6, prove Theorem 36 starting with

$$I = III = (A + \overline{A})(B + \overline{B})(C + \overline{C}).$$

PROBLEM 8. Prove Theorem 18 in a second way, starting with

$$A + B = AI + IB = A(B + \overline{B}) + (A + \overline{A})B.$$

PROBLEM 9. For a further example of the method of only eight distinct columns, begin by computing the following list of sequences, including our eight basic products.

A	0 0 0 0 1 1 1 1
B	0 0 1 1 0 0 1 1
C	0 1 0 1 0 1 0 1
$\bar{A}$	
$\bar{B}$	
$\bar{C}$	
$A\bar{B}$	
$B\bar{C}$	
$C\bar{A}$	
$A\bar{B} + B\bar{C} + C\bar{A}$	
$\bar{A}B$	
$\bar{B}C$	
$\bar{C}A$	
$\bar{A}B + \bar{B}C + \bar{C}A$	
$B + C$	
$A(B + C)$	
AB	
AC	
BC	
$AB + AC + BC$	
$A + B$	
$\overline{AB}$	
$\overline{ABC}$	
$(A + B)\overline{ABC}$	
$AB + (A + B)\overline{ABC}$	
$A\ B\ C$	
$A\ B\ \bar{C}$	
$A\ \bar{B}\ C$	
$A\ \bar{B}\ \bar{C}$	
$\bar{A}\ B\ C$	
$\bar{A}\ B\ \bar{C}$	
$\bar{A}\ \bar{B}\ C$	
$\bar{A}\ \bar{B}\ \bar{C}$	

Use these results to verify Theorems 31 to 36.

5.4 Four Last Theorems.

Finally let's pick up a few results which involve more complicated inversions than we've handled before.

> THEOREM 37. The inverse of any sum is the product of the separate inverses. The inverse of any product is the sum of the separate inverses.

For sums and products of just two sequences this has long since been established in Theorems 11 and 12. Take the case of a double sum, $A + B + C$. Since the order of computation doesn't matter, we can choose to compute $A + B$ first and call the result X. Then

$$\begin{aligned}\overline{A + B + C} = \overline{(A + B) + C} = \overline{X + C} &= \overline{X}\,\overline{C} \\ &= \overline{A + B}\,\overline{C} \\ &= \overline{A}\,\overline{B}\,\overline{C}\end{aligned}$$

As usual, X is a temporary alias. The proof amounts to using Theorem 11 twice. For longer sums the idea is the same; Theorem 11 is used as often as needed. As for products, the action is similar.

$$\begin{aligned}\overline{A\,B\,C} = \overline{(AB)C} = \overline{XC} &= \overline{X} + \overline{C} \\ &= \overline{AB} + \overline{C} \\ &= \overline{A} + \overline{B} + \overline{C}\end{aligned}$$

Theorem 12 has been used twice. For longer products it would be used as many times as needed. Let's put Theorem 37 right to work.

> THEOREM 38. $(A + B + C)\overline{ABC} = A\overline{B} + B\overline{C} + C\overline{A}$.

The proof is a neat bit of teamwork by earlier theory.

$$\begin{aligned}(A + B + C)\overline{ABC} &= (A + B + C)(\overline{A} + \overline{B} + \overline{C}) \\ &= A\overline{B} + A\overline{C} + B\overline{A} + B\overline{C} + C\overline{A} + C\overline{B} \\ &= (A\overline{B} + B\overline{C} + C\overline{A}) + (\overline{A}B + \overline{B}C + \overline{C}A)\end{aligned}$$

But the two double sums in parentheses are equal, by Theorem 34. If we call both of them X for a moment, and remember that $X + X = X$, then the whole side collapses to $A\overline{B} + B\overline{C} + C\overline{A}$, and Theorem 38 is proved. Here is a final pair.

THEOREM 39. $ABC + (A + B + C)\overline{AB + BC + AC}$
$= ABC + A\,\bar{B}\,\bar{C} + \bar{A}\,B\,\bar{C} + \bar{A}\,\bar{B}\,C.$

THEOREM 40. $[(A + B)\overline{AB} + C]\overline{(A + B)ABC}$
$= ABC + A\,\bar{B}\,\bar{C} + \bar{A}\,B\,\bar{C} + \bar{A}\,\bar{B}\,C.$

The proofs will be left as problems, for those hardy souls who have survived to this point, which is as far as we'll penetrate into the theory of sequence arithmetic. There's much more in the literature of mathematics, but we have all we'll need.

PROBLEM 10. Prove Theorems 39 and 40, either by the method of eight distinct columns or otherwise.

PROBLEM 11. Can you complete the following array of sixteen basic products which would be involved in problems of four sequences of A, B, C and D ?

$A\,B\,C\,D$			$\bar{A}\,B\,\bar{C}\,\bar{D}$
$A\,B\,C\,\bar{D}$	$A\,B\,\bar{C}\,\bar{D}$		
$A\,B\,\bar{C}\,D$	$A\,\bar{B}\,C\,\bar{D}$		
$A\,\bar{B}\,C\,D$		$A\,\bar{B}\,\bar{C}\,\bar{D}$	$\bar{A}\,\bar{B}\,\bar{C}\,\bar{D}$

5.5 Abstract Mathematics.

We've come a long way from the basic rules of sequence arithmetic. You now know quite a few fine points of the game, forty theorems' worth, so you're no longer an amateur player. Moreover, play has been honest; we've sometimes labored hard just to 'prove the obvious,' to be sure that we stay on honest, logical ground. So the first two objectives announced in Chapter 1 have had a turn. Part I has concentrated on those two objectives. It has been an example of what is called pure or abstract mathematics. In Part II we'll turn to our third objective. The subject will become applied mathematics, and you'll see that sequence arithmetic is a very useful game. In closing up this first part, a few final questions may be appropriate.

Q. What is abstract mathematics?

A. It's a collection of games. Sequence arithmetic is one of those games.

Q. In sequence arithmetic what do 0, 1, +, ×, and - mean?

A. Nothing yet, but see Part II.

Q. Does $1 + 1$ really equal 1 ?

A. Yes, in sequence arithmetic it really does.

Q. Doesn't that contradict $1 + 1 = 2$?

A. No, because $1 + 1 = 2$ is part of a different game. You expect different games to have different rules.

Q. Are our theorems really true? Lots of them look peculiar.

A. Let's just say that our theorems are provable. Provability is a sort of relative truth. Our theorems are provable from the basic rules, so they're just as 'true' as the basic rules are. Provability is the only kind of truth that concerns the abstract mathematician.

Q. Does sequence arithmetic have any other name?

A. It is often called a Boolean arithmetic. George Boole was one of the early developers of the game.

PART TWO: ABSTRACTION

Chapter 6

FIRST APPLICATION: STATEMENTS

6.1 Sequences as Truth Tables.

Our first application will show you why sequence arithmetic has often been called the 'arithmetic of logic.' It deals with statements. Everyone knows pretty much what a statement is, but here we're going to take a fairly narrow view, so let's start with an example that shows what we worry about and what we don't. Here is a statement.

'My dog has fleas.'

This statement is brief, grammatical and frank English prose, but the only thing that is going to concern us is whether the statement is true or false. Other things such as spelling and punctuation won't matter. True or false is all that counts. But isn't it possible that my dog may have fleas at one time and not at another? Maybe he gets a bath, or perhaps a more interesting host passes by and the fleas migrate. Suppose that my dog gets a daily inspection for one week. The results are recorded; t, for true, means he has fleas.

t t t f t f t

Clearly this is a popular dog. The important point is, however, that in this row of trues and falses we have a sequence. This sequence doesn't use zeros and ones; it uses t's and f's. But putting 1 in place of t and 0 in place of false, the sequence becomes

1 1 1 0 1 0 1

and its meaning is still perfectly clear. The dog is free of fleas only at the Wednesday and Friday inspections. Since our only interest in statements is whether they are true or false, this sequence of zeros and ones contains all we want to know about the statement

'My dog has fleas.'

Essentially it's the sequence, not the English prose, that we need. The sequence of true and false values that goes with a statement will be called the truth table of that statement.

With only one statement to play with, the action in this chapter would be limited, so here is a larger supply; four statements named A, B, C and D.

A: My aunt has fleas.

B: My brother has fleas.

C: My cat has fleas.

D: My dog has fleas.

Fleas will be with us throughout the chapter so relax and enjoy them. To set us up for some later computations, suppose that inspections carried out on the same seven days referred to above produce the following truth tables.

A: 0 0 1 0 1 0 0

B: 0 0 1 0 0 0 1

C: 0 1 1 0 0 0 1

D: 1 1 1 0 1 0 1

There are obvious implications but let's leave them to you to deduce. Notice that the same letter is being used to represent both a statement and its truth table. That's because, as far as we're going to be concerned, the two are inseparable. Two special statements called $\emptyset$ and I will also play roles. Let's introduce these simply by their truth tables,

$\emptyset$ 0 0 0 0 0 0 0

I 1 1 1 1 1 1 1

which makes $\emptyset$ a statement that is always false and I a statement that is always true (on these same seven days). You can translate $\emptyset$ and I into suitable English prose if you want to.

6.2 Or, And, Not.

Next we ask in what ways statements can be combined to make more complicated statements. One obvious example is,

'My aunt has fleas or my brother has fleas.'

Here two of our statements have been connected by or. One or the other of these relatives is in trouble, maybe even both. The possibility that both may have fleas indicates that the word or plays a double role in language. Sometimes it means 'either or both' and sometimes 'one or the other but not both.' The usual decision at this point is to use an inclusive or. This means that when two statements are connected by or, the combined statement is considered true when either or both of the parts are true. In our case, referring back to the sequences A and B, the combined statement then has this truth table.

A + B: 0 0 1 0 1 0 1

The only way for the combined statement to be false is for both of its parts to be false, and that seems to happen here more often than not. It takes only a moment to realize that or is now playing that same role that addition plays in sequence arithmetic. If we add the sequences A and B by the first basic rule of sequence arithmetic, making

$$\begin{array}{cccc} 0 & 0 & 1 & 1 \\ \underline{+0} & \underline{+1} & \underline{+0} & \underline{+1} \\ 0 & 1 & 1 & 1 \end{array}$$

then we get correct true-false values for an inclusive or combination of statements. (The only way to get a 0 is from 0 + 0.) For this reason our latest statement and its truth table will be called A + B.

A + B: My aunt has fleas or my brother has fleas.

You can begin to see an 'arithmetic of statements' developing, with + meaning or.

You can also probably guess what comes next. Another familiar way of combining statements uses and. For example,

'My aunt has fleas and my brother has fleas.'

Common usage tells us that the combined statement is true only when both parts are true. In our case, referring back to the sequences A and B, the combined statement has this truth table.

AB: 0 0 1 0 0 0 0

Only on Tuesday did both people have fleas. Surely this is reminiscent of sequence arithmetic, with and playing the role of multiplication. If we multiply the sequences A and B by the second basic rule of sequence arithmetic, making

$$\begin{array}{cccc} 0 & 0 & 1 & 1 \\ \underline{\times 0} & \underline{\times 1} & \underline{\times 0} & \underline{\times 1} \\ 0 & 0 & 0 & 1 \end{array}$$

then we get correct true-false values for an and combination of statements. For this reason our latest statement and its truth table will be called AB.

AB: My aunt has fleas and my brother has fleas.

The arithmetic of statements is growing. The next step is to consider a statement such as

'My aunt does not have fleas.'

4

Referring back once more to the sequence A, it's easy to produce the truth table that we need here.

$\overline{A}$: 1 1 0 1 0 1 1

It's impossible to escape noticing that not has the same effect on a truth table that inversion has in sequence arithmetic. If we invert a sequence by the third basic rule, replacing 1 by 0 and 0 by 1, then we do get the correct truth table for our latest statement. For this reason the statement will be called $\overline{A}$.

$\overline{A}$: My aunt does not have fleas.

The three operations of sequence arithmetic have now been translated into the language of statements.

PROBLEM 1. Translate these symbols to English prose,

CD:

C + D:

$\overline{C}$:

and this English prose to the symbols of sequence arithmetic.

: My aunt has fleas and my cat has fleas.

: My brother does not have fleas.

: My aunt has fleas or my cat has fleas.

6.3 Translation.

Moving on to slightly more interesting translations, how does

$A\overline{B}$:

sound in English? If you come up with 'My aunt has fleas, but my brother does not' then you're right. The word but plays the same role as and in places like this, at least in so far as truth or falseness is concerned. Try another.

ABC:

A direct, but wordy, translation is 'My aunt has fleas and my brother has fleas and my cat has fleas.' But surely anyone would shorten that to 'My aunt and brother and cat all have fleas.' Here are some more abbreviated but accurate translations.

$\bar{A} + \bar{B}$: Either my aunt doesn't have fleas or my brother doesn't.

$\overline{A + B}$: It isn't true that one or both of them has fleas.

$\bar{A}\,\bar{B}CD$: Only the cat and the dog have fleas.

Examine them carefully to see if you agree.

To discover when one of these statements is true and when false, one systematic way is to use the symbolic translation. Take the last statement as an example. From the truth tables A, B, C and D the computation of $\bar{A}\,\bar{B}CD$ by the rules of sequence arithmetic is routine.

$$\bar{A}\,\bar{B}CD \qquad 0\ 1\ 0\ 0\ 0\ 0\ 0$$

So 'Only the cat and the dog have fleas' is true just on Monday. For the next to last statement the truth table is

$$\overline{A + B} \qquad 1\ 1\ 0\ 1\ 0\ 1\ 0$$

and so 'It isn't true that one or both of them has fleas' is false on Tuesday, Thursday and Saturday.

PROBLEM 2. Match these statements with the symbols which follow by writing one of the letters S to Z before each statement.

: All four have fleas.

: It isn't true that all four have fleas.

: At least one of them has fleas.

: Only the dog has fleas.

: Exactly one of them has fleas.

: My aunt has fleas, and so does either my brother or my cat or my dog.

: None of the four has fleas.

: Either my aunt or my brother has fleas, and so does either the cat or the dog.

$S = \overline{ABCD}$	$W = \bar{A}\,\bar{B}\,\bar{C}\,\bar{D}$
$T = A(B + C + D)$	$X = A + B + C + D$
$U = (A + B)(C + D)$	$Y = \bar{A}\,\bar{B}\,\bar{C}\,D$
$V = ABCD$	$Z = \bar{A}\,\bar{B}\,\bar{C}\,D + \bar{A}\,\bar{B}\,C\,\bar{D} + \bar{A}\,B\,\bar{C}\,\bar{D} + A\,\bar{B}\,\bar{C}\,\bar{D}$

PROBLEM 3. When is the statement 'It isn't true that they all have fleas' false?

PROBLEM 4. When is the statement 'Either my aunt or my brother has fleas, and so does either my cat or my dog' true?

6.4 Simplification.

Now that you've had a little practice at translating back and forth between the languages of English and sequence arithmetic, let's apply the strategy, or theory, that we've worked out in earlier chapters. As a first example take the redundant

'My aunt has fleas or my aunt has fleas.'

which clearly translates to

$$A + A.$$

Since we know that $A + A = A$, our redundant statement simplifies to

'My aunt has fleas.'

Of course, in this staggeringly simple example the strategy of sequence arithmetic is more a luxury than a necessity. A toddler could make this simplification without leaving the field of English prose. One important point bears repeating, however. The original redundant statement and its simplification are clearly not identical. They differ in length, punctuation, and many other ways. But, and this is the point, both have identical truth tables. As far as truth or falseness goes, there's no difference between them. Here's a second easy example.

'It isn't true that my aunt does not have fleas.'

After spotting the double negative you'll surely decide that this is just

'My aunt has fleas.'

all over again, just by optical inspection. But what is the symbol for this redundant statement? Isn't it just $\bar{\bar{A}}$? And don't we have a theorem which guarantees that $\bar{\bar{A}} = A$? This theorem is again a luxury rather than a necessity for this particular problem, but at least theory and common sense both lead us back to statement A.

Graduating now to some slightly more challenging examples, consider the statement

'My aunt and brother both have fleas, or else
she does but he doesn't.'

Perhaps you can see at once how that statement can be simplified. But whether you can or not, here is how it translates,

$$AB + A\overline{B}$$

which will strike a familiar note. According to Theorem 15, such a sequence is identical with A itself,

$$AB + A\overline{B} = A$$

in so far as truth is concerned. So once again we can simplify what we have to

'My aunt has fleas.'

Next compare these two statements.

$\overline{A}\ \overline{B}$: Neither my aunt nor my brother has fleas.

$\overline{A + B}$: It isn't true that one or both of them does.

You can choose between them strictly on the basis of brevity or clarity or personal taste, because our old result about the inverse of a sum and the product of the inverses (Theorem 11) shows they have identical truth tables. The same is true of this pair.

$(A + B)(C + D)$: Either my aunt or brother has fleas, and so does either my cat or my dog.

$AC + AD + BC + BD$: My aunt and cat both have fleas, or else she and the dog do, or maybe it's my brother and cat, or maybe my brother and dog.

This time Theorem 30 comes to our rescue. The English prose is getting more complex, and translation into the symbols of sequence arithmetic begins to look more like a necessity than a luxury. Our theorems can be a help in comparing and simplifying complex statements. Of the above pair $(A + B)(C + D)$ seems both simpler and clearer.

PROBLEM 5. Write either A, I or $\emptyset$ on the lines provided to show what the following statements can be reduced to.

_____ Either my aunt does have fleas or else she doesn't.

_____ My aunt has fleas and she doesn't have fleas.

_____ My aunt has fleas and my aunt has fleas.

_____ Either my aunt has fleas or else she and my brother both do.

PROBLEM 6. Do any of these statements have the same truth table?

a) Either my aunt or my brother has fleas, and one or the other doesn't.

b) Either my aunt has fleas and my brother doesn't, or vice versa.

c) Either my aunt or my brother has fleas, but it isn't true that they both do.

PROBLEM 7. Do these two statements have the same truth table or not?

a) My aunt has fleas but my brother does not, or else he does but the cat doesn't, or maybe the cat does and my aunt doesn't.

b) My aunt doesn't have fleas but my brother does, or else he doesn't but the cat does, or maybe the cat doesn't and my aunt does.

6.5 The Lady or the Tiger.

For a final example let's get away from fleas and puzzle out this ancient dilemma. A captured warrior, the prince of his tribe, is given the following sporting chance by the chief of his captors. 'You see these two doors. Behind the one is my daughter, behind the other a hungry tiger. I shall have either one of these doors opened, whichever you choose. To help you I will permit you to put one statement to one of these two guards. He will answer simply true or false. However, I warn you that one of these guards never speaks the truth, whereas the other never lies.'

What statement should the warrior make? At first glance his chances look about fifty-fifty, but every student of logic has thought his way through at least one old chestnut of this sort, so watch how the apparent fifty-fifty can be turned into a sure thing. There are two basic statements with which this warrior is concerned. First, pointing to one of the two doors, he could say,

A: This is the lady's door.

(Notice that A now represents a different statement; it doesn't refer to aunts and fleas anymore.) The other basic statement is, pointing to one of the two guards,

B: You tell the truth.

Our warrior's problem is that he doesn't know whether these important statements are true or false. If it occurs to him to try all the various combinations of true and false, then he might be led to consider these truth tables for A and B.

A: 0 0 1 1

B: 0 1 0 1

There are four possible combinations, and each of these four short columns displays one of those combinations. In column one, both statements are false; in columns two and three, one is true and one false; in column four, both are true. Put in another way, our warrior's problem is that he won't know which of the four columns he's picked, when he chooses one door and one guard. Eventually, it might occur to him that it would be awfully nice if he could arrange for the guard's answer to be as follows.

Guard's answer: 0 0 1 1

Call it wishful thinking if you want, but if our warrior could formulate a statement which would bring these replies (depending on which door and which guard he points to) then his problem would be solved. Because this truth table is identical with that of A, so that a reply of 'false' comes just when A is false, and a reply of 'true' comes just when A is true. He can believe the answer he gets, at least for distinguishing doors. (He still won't know whether it was the truth-teller or the liar who answered, but presumably he doesn't care.)

But what statement can possibly achieve this miracle? Take the four columns one by one. In the first, the guard's answer is to be 'false.' But in this column B happens to be false, so it's the liar who is giving this answer. If the liar says 'false,' then the warrior's statement would have to be true, so the truth table for the (still unknown) warrior's statement would have to lead off with a 1. In the second column the guard's answer is still 'false.' But here B is true, so it's the truth-teller speaking. And if the truth-teller says 'false,' then the warrior's statement would have to be false. Argue it out for the remaining two columns yourself. The truth table for the warrior's statement would have to be this. Do you agree?

Warrior's statement: 1 0 0 1

So now we face the job of producing a statement which has a specified truth table. If this brings recollections of 'the method of only four distinct columns' that got heavy use in Chapter 4, then you're not far from the finish. One statement which has such a truth table is $AB + \bar{A}\,\bar{B}$.

Check that by the usual rules of sequence arithmetic if you have to, but this is a suitable miracle statement for our warrior. Translating it into English prose, we have 'This is the lady's door and you tell the truth, or else it is not the lady's door and you lie.' If the guard fully understands this statement, our man is safe. At least, he can have the lady instead of the tiger.

If you enjoy this sort of thing, here is an easier one that you can figure out in your head. An explorer is in a region inhabited by two tribes. The members of one tribe always lie, the members of the other always tell the truth. He meets two natives. 'Are you a truth-teller?' he asks the tall one. 'Goom,' the native replies. 'He says, yes,' explains the short native, who speaks English. 'Him big liar.' Which tribe did each belong to? I doubt that you'll need any sequence arithmetic to find the answer.

6.6 Summary.

The examples of this chapter have obviously been light-hearted. Their purpose has been to show the close relationship between sequences and statements, and to suggest how sequence arithmetic can be helpful in unsnarling the logic of complex statements. It will no doubt be a long time before courtroom trials and legislative affairs are conducted in symbolic language but the possibility is not beyond the realm of science fiction.

Notice particularly that, in this application, the ideas of sequence arithmetic pick up a kind of 'meaning,' 0 meaning false, 1 meaning true, + meaning <u>or</u>, and so on. And the mysterious $1 + 1 = 1$ means simply that when statements A and B are both true, then so is $A + B$. It amounts to our decision to use an inclusive or. If we had voted for 'either one or the other but <u>not both</u>,' then $1 + 1$ would be 0 instead of 1. So the mystery solved, and as so often happens, the solution proves to be extremely simple. For two altogether different-looking solutions, however, see Chapters 7 and 8.

Chapter 7

SECOND APPLICATION: SUBSETS

7.1 Sequences as Membership Lists.

Suppose there are just ten prisoners in a small jail. The following table shows which of the ten belong to certain groups, or to use the official term to certain subsets. For convenience the subsets have been named A, B, C and D.

		1	2	3	4	5	6	7	8	9	10
(Redheads)	A:	0	1	1	0	0	1	0	0	0	1
(First offenders)	B:	0	0	1	1	1	1	1	1	0	0
(Six-footers)	C:	1	0	0	0	1	1	0	0	0	1
(Females)	D:	0	0	0	1	0	1	1	0	0	0

The symbol 1 means that a prisoner belongs, and 0 means he doesn't. Prisoners 2, 3, 6 and 10 are redheads, the others are not. Prisoners 3 to 8 are first offenders, and so on. Each row of this table amounts to a membership list for that particular subset. Obviously each row is also a sequence of 0's and 1's, and we are going to use the letters A, B, C and D to represent these sequences. So these letters will be doing double duty, representing both the subsets and the sequences, but this won't cause us any trouble. The sequence is the membership list that tells who belongs to the subset and who doesn't, and this question of membership is the only thing about the subset that will concern us. Lots of other subsets can be imagined, and the appropriate membership lists could be worked out from the jail records, but two special subsets deserve special mention.

	1	2	3	4	5	6	7	8	9	10
I:	1	1	1	1	1	1	1	1	1	1
$\emptyset$:	0	0	0	0	0	0	0	0	0	0

Subset I includes all the prisoners. It is the master subset (or set) from which the other subsets are drawn. It may seem odd to call it a subset, but it does no harm. Subset $\emptyset$ is called the 'empty subset' and it's easy to see why. It has no members at all. It may also seem odd to call this a subset, but it's customary and useful.

PROBLEM 1. What can you say about prisoner 6 ? What about prisoner 9 ?

7.2 Union, Intersection and Complement.

Which prisoners are either redheaded or first offenders? To put the same question in another way, if we merge the subsets A and B into a single subset, who belongs and who doesn't? Here is the membership list which provides the answer.

(Redhead or first offender) 0 1 1 1 1 1 1 1 0 1

Check it yourself. Prisoners 1 and 9 are the only ones who fail to qualify. This new subset is less exclusive than either A or B. To be admitted to the merger it's enough to belong to A or to B or to both. The only way to fail is to be in neither A nor B, like prisoners 1 and 9. I hope this reminds you of the addition process of sequence arithmetic. If we add sequences A and B we get exactly the membership list for our merger.

A + B: 0 1 1 1 1 1 1 1 0 1

And this is no coincidence. The only time we want a 0 here is when A and B have matching 0's. And that is precisely what sequence addition offers us. The merger of two subsets A and B is called their union. We represent it by A + B. Next, which prisoners are both redheaded and first offenders? Here is the membership list which provides the answer.

(Redheaded first offender) 0 0 1 0 0 1 0 0 0 0

Only prisoners 3 and 6 qualify. This new subset is more exclusive than either A or B. To be admitted you must belong to both A and B, and that will remind you of the multiplication process of sequence arithmetic. If we multiply A and B we get exactly the membership list for our latest subset.

AB: 0 0 1 0 0 1 0 0 0 0

And this is no coincidence. The only time we want a 1 here is when A and B have matching 1's. And that is precisely what sequence multiplication offers us. This new subset is called the intersection of A and B. It is also known as their overlap, since it picks out the common members of both. The intersection of A and B is represented by AB. One operation of sequence arithmetic is left, inversion, and its application to subsets isn't hard to guess. Which prisoners are not redheaded? Here is the appropriate membership list.

(Not redheaded) 1 0 0 1 1 0 1 1 1 0

To belong here a prisoner must not belong to subset A, and vice versa. This leads to a complete reversal of the 0's and 1's. Inverting the A sequence certainly brings this same result.

$\bar{A}$: 1 0 0 1 1 0 1 1 1 0

Our new subset is called the <u>complement</u> of A, or sometimes the inverse. It is represented by $\bar{A}$.

And so we come to the conclusion that whenever subsets are merged, intersected or complemented, their membership lists are added, multiplied or inverted according to the rules of sequence arithmetic. A few more examples will be enough to make this crystal clear. The subset of prisoners who are redheaded, but not first offenders (in A but not in B), has this membership list.

$A\bar{B}$: 0 1 0 0 0 0 0 0 0 1

Only prisoners 2 and 10 qualify. The subset of prisoners who are redheaded, not first offenders, and female (in A, not in B, in D) has this membership list.

$A\bar{B}D$: 0 0 0 0 0 0 0 0 0 0

Nobody qualifies. This subset is empty. If we ask for redheads who are also either first offenders or very tall (in A and also in either B or C) then the list

A(B + C): 0 0 1 0 0 1 0 0 0 1

shows that prisoners 3, 6 and 10 are available. Once a subset has been described in the symbolism of sequence arithmetic the computation of its membership list becomes routine.

PROBLEM 2. Compute membership lists for the following subsets and describe the membership of each. (For example, $A\bar{D}$ includes male redheads.)

$A\bar{D}$:

ABCD:

$A\bar{D} + \bar{A}D$:

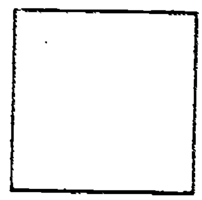
(A + B)(C + D):

$\bar{A}\ \bar{B}$:

$B + \bar{D}$:

7.3 Subset Diagrams.

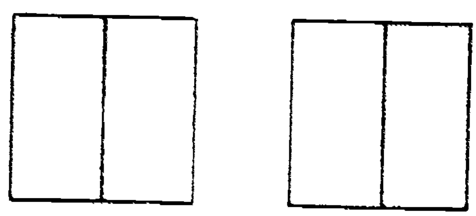
In large scale subset problems the use of sequences as membership lists, and of sequence arithmetic for computing membership lists is neat, natural and necessary. In a wide variety of modern applications, however, only a few subsets see action at any one time. For such problems, instead of identifying subsets by membership lists, we can use an ancient device called the subset diagram. Imagine some master set I, perhaps our ten prisoners or perhaps some much larger set. Also imagine each member of I represented by a number or a spot inside this square.



I

To identify any subset of I we can now draw an interior boundary enclosing the numbers or spots which represent the members of that subset. For instance, if we gather the members of some subset A in the left half of the square, then a vertical boundary down the center encloses the members of A in that half. It also puts the non-members of A (members of $\bar{A}$) into the other half.

A

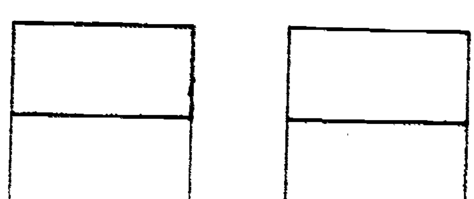


$\bar{A}$

One advantage of subset diagrams already begins to appear. They offer us a clear picture of how various subsets are related to each other. Merging the A and $\bar{A}$ regions, for example, we certainly seem to have the entire square. This is how diagrams illustrate our $A + \bar{A} = I$ theorem. It also seems crystal clear that the A and $\bar{A}$ regions have no overlap, which illustrates $A\bar{A} = \emptyset$.

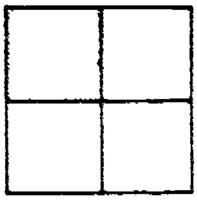
With subsets A and $\bar{A}$ neatly sorted to left and right, suppose we also bring the members of some subset B to the top half of our square. A horizontal boundary through the center encloses B and $\bar{B}$.

B

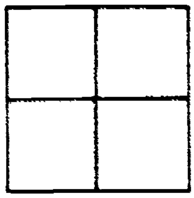


$\bar{B}$

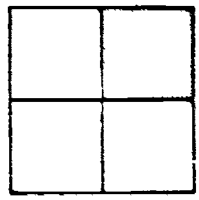
Various unions, intersections and complements now become easy to visualize. Take these four famous intersections first. They will remind you of four basic products.



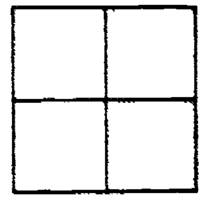
AB



$A\overline{B}$



$\overline{A}B$



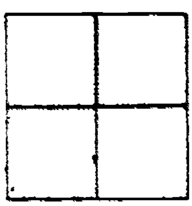
$\overline{A}\ \overline{B}$

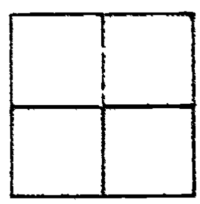
Members of both A and B are in the upper left quarter, where the A and B regions overlap; members of A but not of B are at the lower left, etc. Together these four merge to the master set I, nicely illustrating theorem 20.

$$AB + A\overline{B} + \overline{A}B + \overline{A}\ \overline{B} = I$$

The union of only AB and $A\overline{B}$ produces the left half of the square, which still represents subset A. This illustrates $AB + A\overline{B} = A$. The other familiar unions have these diagrams.

$A + B$

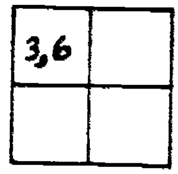




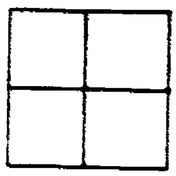
$\overline{A} + \overline{B}$

Since shading of any subset leaves its complement (or inverse) unshaded, and vice versa, we now have a picturesque view of quite a number of theorems. Compare the diagrams of $A + B$ and $\overline{A}\ \overline{B}$, for instance. They are complements, which illustrates $\overline{A + B} = \overline{A}\ \overline{B}$. Or compare the diagrams of A, $\overline{A}B$ and $A + B$. The first two together make the last, illustrating $A + \overline{A}B = A + B$. A few similar results are suggested in the following problems. This sort of picturesque display of unions, intersection and complements is what has made subset diagrams so popular.

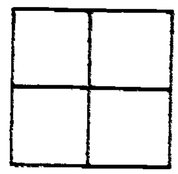
PROBLEM 3. As this diagram shows, prisoners 3 and 6 of our opening example fall in subset AB. Indicate whether the others fall in $A\bar{B}$, $\bar{A}B$ or $\bar{A}\,\bar{B}$ by writing their numbers in the appropriate quarter.



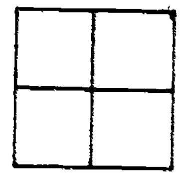
PROBLEM 4. Refer back to the diagrams of this section to properly shade the following.



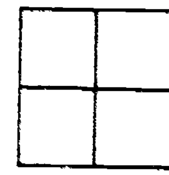
$A\bar{B} + \bar{A}B$



$AB + \bar{A}\,\bar{B}$



$A + \bar{B}$

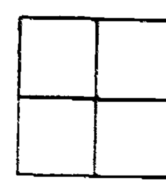


$\bar{A} + B$

PROBLEM 5. Compare the diagrams of AB and $\bar{A} + \bar{B}$. Which theorem does this illustrate?

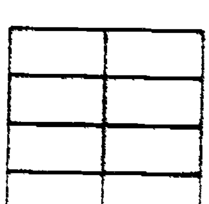
PROBLEM 6. From the diagram of $A + B$ note which three of the four quarters of our square are shaded. What theorem does this illustrate?

PROBLEM 7. From diagrams of $A + B$ and $\bar{A} + \bar{B}$ deduce which parts of our square are in $(A + B)(\bar{A} + \bar{B})$, and shade those parts in the diagram at the right. Which theorem does this illustrate?

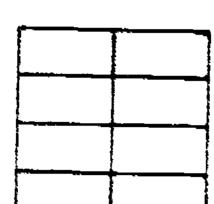


7.4 More Diagrams.

For problems involving three subsets A, B and C a popular procedure is to group the members of C in a belt across the middle of the diagram, splitting each of the four regions we already have into two parts.

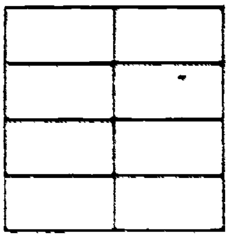


C

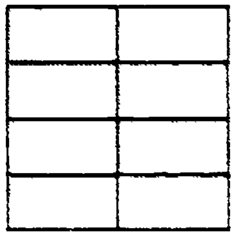


$\bar{C}$

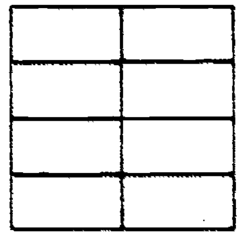
Keeping subset A in the left half of the diagram and subset B in the top half, as before, it isn't difficult to locate various simple unions and intersections. Here are some typical ones, shaded as usual.



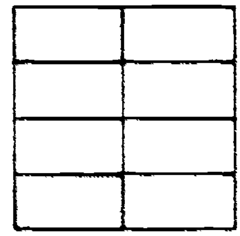
AC



$A + C$



$A + \overline{C}$



$A\overline{C}$

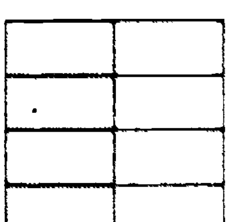
The master set I has now been split into eight parts. The significance of each part is easy to discover, and it will come as only a mild surprise that these eight parts correspond to eight basic products.

$AB\overline{C}$	$\overline{A}B\overline{C}$
ABC	$\overline{A}BC$
$A\overline{B}C$	$\overline{A}\overline{B}C$
$A\overline{B}\overline{C}$	$\overline{A}\overline{B}\overline{C}$

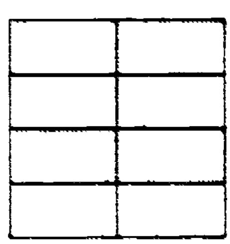
6	2
7	3
5	1
4	0

It becomes tiresome writing these products over and over, so each of these eight subsets is given a number, from 0 to 7, as shown at the right. (The popularity of this unusual number pattern will be explained in Chapter 9.) Notice first that subset 7 is inside A, B and C. This makes it the home of ABC, as shown. Similarly, subset 6 is inside A and B but outside C. This makes it the home of $AB\overline{C}$. Check the other six yourself, to be sure you agree with their product labels.

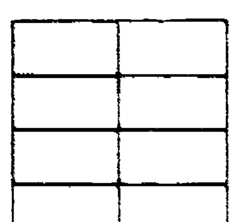
Diagrams of this sort offer us pleasant artist's impressions of our various theorems about A, B and C. Take first A, $B + C$ and their intersection.



A

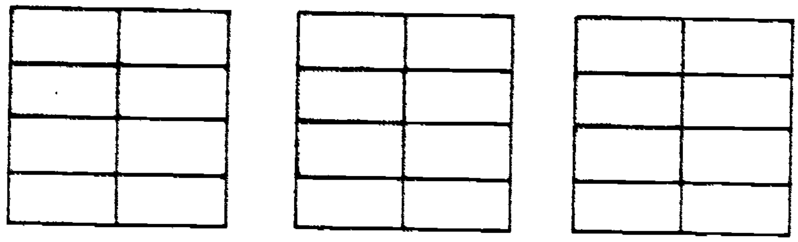


$B + C$



$A(B + C)$

You can see that the subset at the right is the overlap of the first two. Subsets 5, 6 and 7 make up this overlap. Now take AB, AC and their union.



AB AC AB + AC

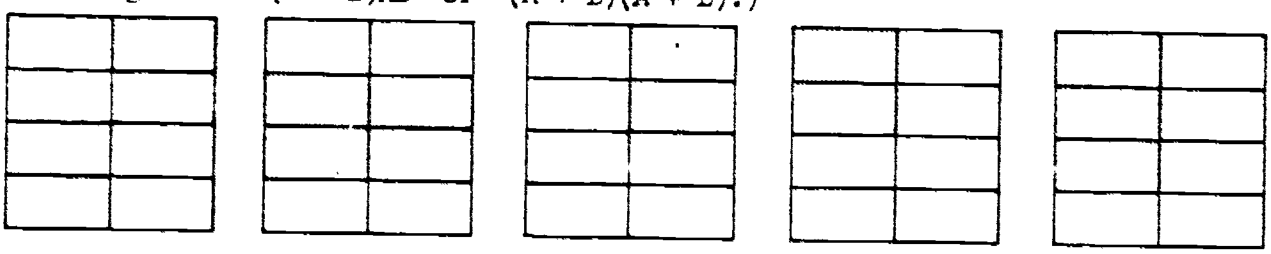
You can see that the subset at the right is the union, or merger, of the first two. Subsets 5, 6 and 7 make up this union. We have come upon 5, 6 and 7 in two different ways and obtained a picturesque view of Theorem 23,

$$A(B + C) = AB + AC$$

as well as of Theorem 31.

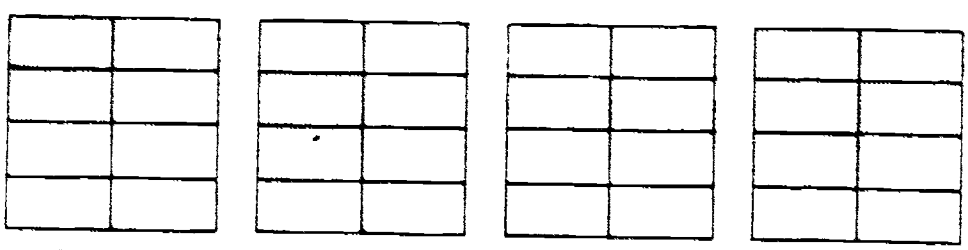
$$A(B + C) = A\overline{B}C + AB\overline{C} + ABC$$

As a second example let's watch $AB + (A + B)\overline{AB}C$ develop. (See Problem 7 for the diagram of $(A + B)\overline{AB}$ or $(A + B)(\overline{A} + \overline{B})$.)



$(A + B)\overline{AB}$ C $(A + B)\overline{AB}C$ AB $AB + (A + B)\overline{AB}C$

Subsets 3, 5, 6 and 7 are included in the finished product. Now watch AB + AC + BC grow.



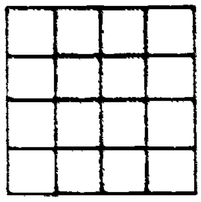
AB AC BC AB + AC + BC

Again it's the subsets 3, 5, 6 and 7 which are included. We have come to these four in two different ways and achieved a picturesque view of Theorems 32 and 33.

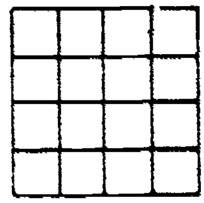
$$AB + AC + BC = AB + (A + B)\overline{AB}C$$

$$AB + AC + BC = \overline{A}BC + A\overline{B}C + AB\overline{C} + ABC$$

The most popular way to achieve a four-subset diagram is by a vertical belt down the center of our diagram, with the members of a fourth subset D inside the belt and the members of $\bar{D}$ outside.

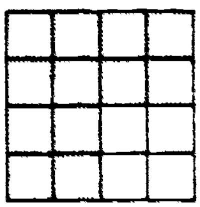


D

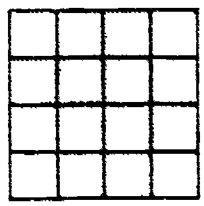


$\bar{D}$

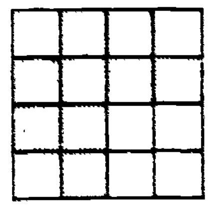
Keeping the areas for A, B and C just as before, it is still easy to locate the various intersections and unions. Watch (A + B)(C + D) develop.



A + B

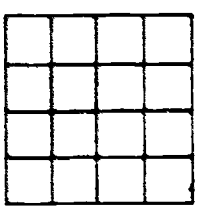


C + D

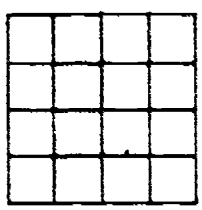


(A + B)(C + D)

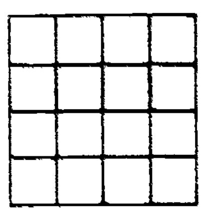
Also watch AC + AD + BC + BD develop.



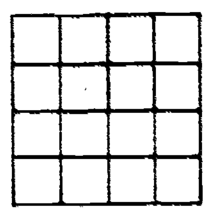
AC



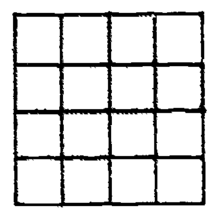
AD



BC



BD



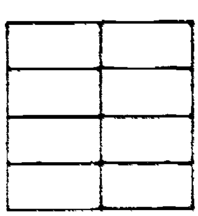
union

Notice that in both these examples the final result is the same, as Theorem 30 guarantees.

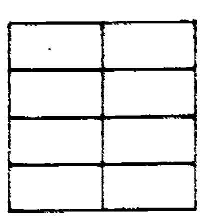
$$(A + B)(C + D) = AC + AD + BC + BD$$

It is possible to develop diagrams for more than four subsets, but they become fairly complicated and we'll leave them to the professionals.

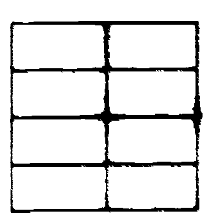
PROBLEM 8. Show that $(A + B + C)\overline{ABC}$ includes six of the eight basic areas by shading each of the following diagrams.



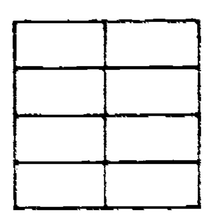
ABC



$\overline{ABC}$



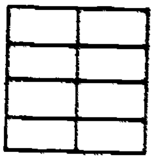
A + B + C



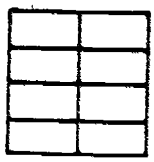
$(A + B + C)\overline{ABC}$

Which of our theorems does this illustrate?

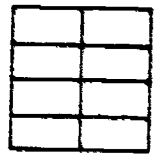
PROBLEM 9. Show that $A\overline{B} + B\overline{C} + C\overline{A}$ and $\overline{A}B + \overline{B}C + \overline{C}A$ both include the same six numbered subsets as in Problem 8.



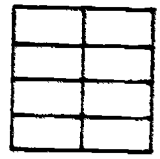
$A\overline{B}$



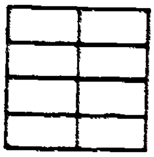
$B\overline{C}$



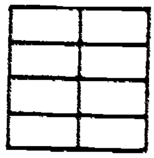
$C\overline{A}$



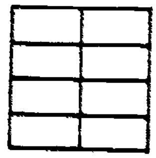
union



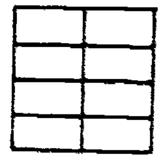
$\overline{A}B$



$\overline{B}C$



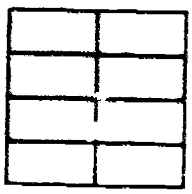
$\overline{C}A$



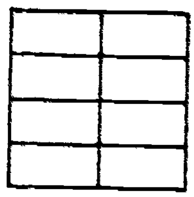
union

Which of our theorems does this illustrate?

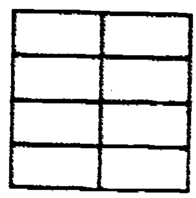
PROBLEM 10. Show that $ABC + (A + B + C)\overline{AB + AC + BC}$ includes four of the eight basic areas. (Refer back for diagrams of $AB + AC + BC$ and $A + B + C$.)



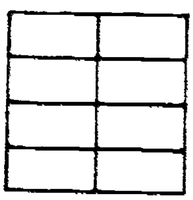
$AB + AC + BC$



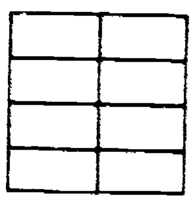
$\overline{AB + AC + BC}$



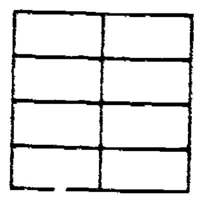
$A + B + C$



$(A+B+C)\overline{AB+AC+BC}$



ABC



final

Which of our theorems does this illustrate?

PROBLEM 11. One of the following two subsets is identical with the subset of Problem 10. Use subset diagrams to discover which one it is.

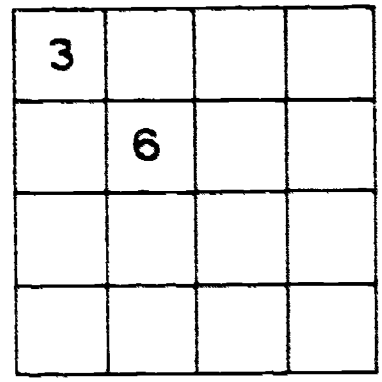
a) $[(A + B)\overline{AB} + C](A + B)\overline{AB}C$

b) $[(A + B)\overline{AB} + C]\overline{A + B}\ \overline{AB}C$

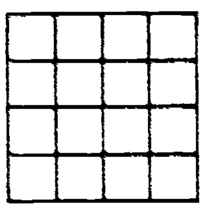
c) $[(A + B)\overline{AB} + C]\overline{(A + B)\overline{AB}C}$

6

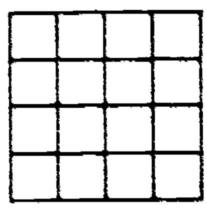
PROBLEM 12. Each of the sixteen parts of this diagram corresponds to one of sixteen basic products. Prisoner 3 (of our opening example) falls into subset $A\ B\ \overline{C}\ \overline{D}$, and prisoner 6 into subset ABCD, as shown. Write the number of each of the other prisoners in the appropriate part. Which basic product represents that part of the diagram?



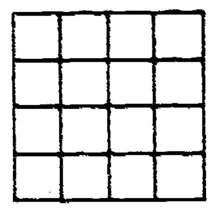
PROBLEM 13. Show that $A(B + C + D)$ includes seven of the sixteen parts by shading each of these diagrams.



A

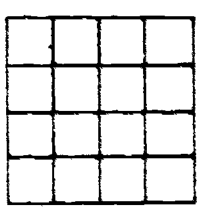


B + C + D

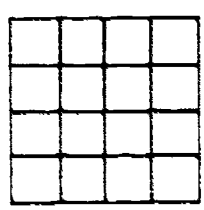


A(B + C + D)

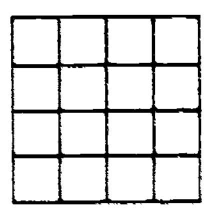
Also show that $AB + AC + AD$ includes exactly the same seven parts.



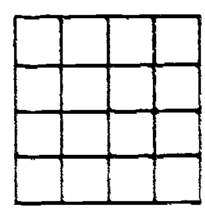
AB



AC

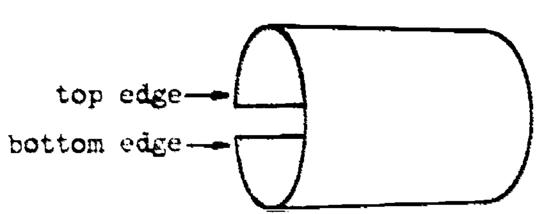


AD



AB + AC + AD

PROBLEM 14. A possible objection to the horizontal and vertical belts we have used to represent subsets C and D is that they leave $\overline{C}$ and $\overline{D}$ disconnected. (Each consists of two separate strips.) This is not a very important objection. Major users of subset diagrams don't mind the disconnectedness. Perhaps the most amusing way to connect up the separate parts of $\overline{C}$ is to convert our square into a cylinder by bringing its top and bottom edges together.



PROBLEM 14. (Continued)

Subset C can occupy the front face of the cylinder and $\bar{C}$ the rear face. Where will A, B, D and their inverses end up? This still leaves $\bar{D}$ disconnected. How can its two parts be brought together?

7.5 A Mouse-Maze Problem.

The main purpose of this chapter has been to show that certain problems involving subsets are applications of sequence arithmetic, with the ideas of union, intersection, and complement playing the roles of addition, multiplication and inversion. When only a few subsets are involved at one time the relationships between them are neatly exhibited by subset diagrams, with the result that sequences (playing their roles of membership lists) seldom see action. In more complicated problems, with many subsets involved, diagrams become more confusing and it is probably wise to work directly with the sequences themselves. In such cases calculating machines will probably be used for the computations. Machines handle the longest sequences with relative ease.

The idea of subset finds a place in almost all parts of mathematics, so we'll close up this chapter with two typical small-scale examples. Suppose twenty mice are introduced one-by-one into a maze. If a mouse comes out the correct exit he is rewarded with a piece of cheese; otherwise he gets nothing. Each mouse makes three tries. With A, B and C representing the subsets that make successful first, second and third tries, respectively, the results are as shown here at the left. (At the right is a reminder of our basic pattern.)

0	1
3	3
1	4
0	8

$A\ B\ \bar{C}$	$\bar{A}\ B\ \bar{C}$
$A\ B\ C$	$\bar{A}\ B\ C$
$A\ \bar{B}\ C$	$\bar{A}\ \bar{B}\ C$
$A\ \bar{B}\ \bar{C}$	$\bar{A}\ \bar{B}\ \bar{C}$

The diagram provides a simple summary of the results and can be used to answer numerous questions about the experiment.

Question: How many mice were successful every time?
Answer: Look in ABC. Three mice.

Question: How many were never successful?
Answer: Look in $\bar{A}\,\bar{B}\,\bar{C}$. Eight mice.

Question: How many didn't make it until the final try?
Answer: Look in $\bar{A}\,\bar{B}\,C$. Four mice.

Question: How many made it on the very first try?
Answer: Subset A occupies the entire left half. Four mice.

Question: How many made it on the second try?
Answer:

Question: How many on the third try?
Answer:

Question: How many mice had at least one success?
Answer:

Question: How many times was a success followed by a success?
Answer:

Question: How many times was a success followed by a failure?
Answer:

This example borders on the subject of probability in which subsets play a conspicuous part.

7.6 <u>A Problem of Detection</u>.

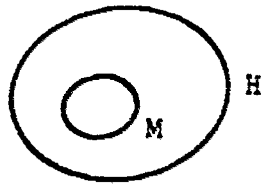
Suppose that the following somewhat improbable facts are true.

a) The murderer wore a tall silk hat.
b) All Irishmen are redheaded.
c) The butler's name is O'Brien.
d) Redheads never wear hats.

What can be deduced? Presumably a good detective would unravel the facts in no time at all, but as a last example of subsets let's stretch things out a bit, using these subsets.

E. Irishmen
B. The butler
R. Redheads
H. Hat-wearers
M. The murderer

Subsets B and M have only one member each, and our problem is to discover whether or not B and M are the same. Fact (a) puts subset M inside of subset H. For situations of this sort a slight variation of our subset diagram is more convenient. Instead of dividing up our square in the now familiar patterns, we just draw loops surrounding the members of each subset. To show that M is completely inside H, we simply put the M loop completely inside the H loop.



Take fact (d) next. It guarantees there is no overlap between redheads and hat-wearers. So subsets R and H must not overlap on our diagram, which now grows to this.

Coming to fact (b) we find that the whole of E is inside R.

Finally, fact (c) puts the butler somewhere inside E.

The diagram now shows clearly that B and M are not the same man, which was probably crystal clear in the first place. It may not be current practice to use subset diagrams in criminal investigations, but you can at least see the

possibilities. In a complicated case it may even be useful to bring membership lists into action, and let computers do the detective work.

PROBLEM 16. Assuming that

a) no one who is going to a party ever fails to brush his hair,
b) no one looks fascinating, if he is untidy,
c) opium eaters have no self-command,
d) everyone who has brushed his hair looks fascinating,
e) no one wears white kid gloves unless he is going to a party,
f) a man is always untidy, if he has no self-command,

make a deduction which uses all these facts. (This is one of many such problems created by Lewis Carroll, the author of <u>Alice in Wonderland</u>.)

7.7 <u>Summary</u>.

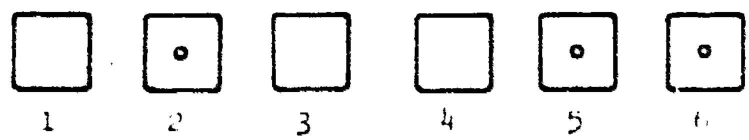
This chapter, like the last, has been intentionally light-hearted. It has tried to show how sequence arithmetic is related to subset affairs. Notice particularly that the 1's and 0's of our sequences have again been assigned a 'meaning.' In our first application they meant <u>true</u> and <u>false</u>, and sequences were truth tables. Now 1 and 0 mean <u>member</u> and <u>non-member</u>, and sequences are membership lists. And the mysterious $1 + 1 = 1$ now means merely that being in A and also in B surely puts you in the union $A + B$. Things that look so strange in abstract mathematics can look so simple in the application.

Chapter 8

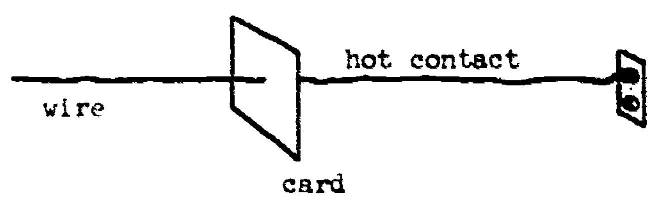
THIRD APPLICATION: SIGNALS

8.1 Sequences as Signals.

Now we turn to an application that will be given more extensive coverage. Not that statements and subsets aren't important, but this third application will get the lion's share of our attention because of a surprising twist that it takes. To begin, imagine that these six cards



are slipped, one by one, between a wire and a hot electrical contact.



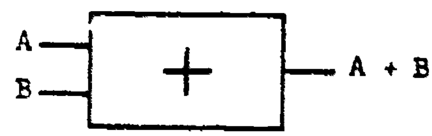
Cards 2, 5 and 6 have a hole punched in them, the others do not. So if things are lined up accurately, the wire will touch the hot contact only when cards 2, 5 and 6 are in position. Contact will be made through the holes, and current will flow in the wire. When cards 1, 3 and 4 are in position there is no hole, no contact, and no current. As the six cards are slipped successively into position, the electrical experiences of our wire can be summarized in this way,

$$0\ 1\ 0\ 0\ 1\ 1$$

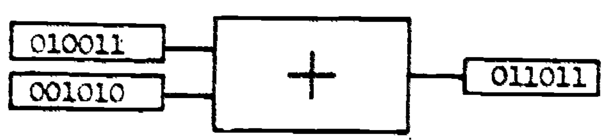
where 0 stands for a cold wire (no current flow) and 1 for a hot one. Our electrical apparatus amounts to a sort of 'card reader.' It converts a sequence of cards, with holes or without, into a sequence of electrical hots and colds. The wire carries a sort of electrical signal, and our sequence of zeros and ones is a record of that signal. In the chapter this is the sort of role that sequences will play. They will represent electrical signals.

8.2 The +, × and - boxes.

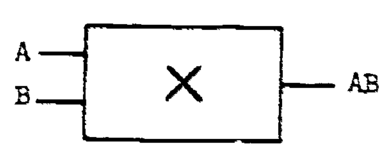
The simplicity of converting sequences to electrical form has led to the invention of devices for adding, multiplying and inverting sequences, according to the basic rules of sequence arithmetic. For example, a 'plus box' will take any two electrical sequences as 'inputs' (call these input sequences A and B) and produce the correct sequence A + B as 'output.'



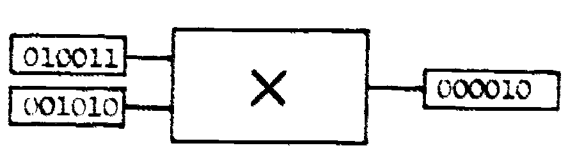
It takes each pair of values of the input sequences in turn and generates the correct output value. In the following example the successive steps are 0 + 0 = 0, 1 + 0 = 1, 0 + 1 = 1, and so on.



The plus box is designed to perform this familiar addition process. It produces a cold output only when both input contacts are cold. How it does this is not something we'll go into here, but a few hints are offered in Problem 1. There is also a device which we'll call a 'times box.' It takes two input sequences, A and B, and generates the product sequence AB, value by value.



For the same input sequences used above, the successive steps are 0 × 0 = 0, 1 × 0 = 0, 0 × 1 = 0, and so on.

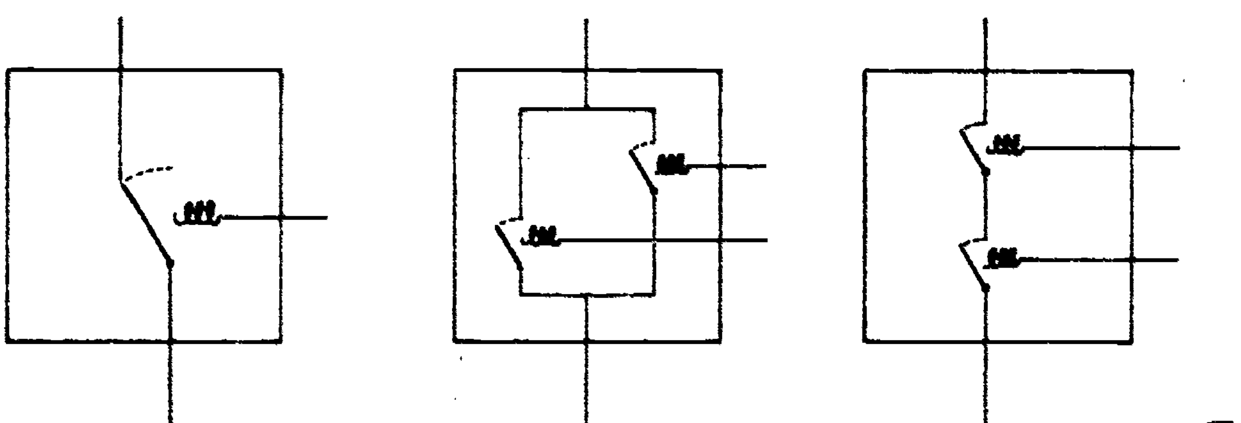


The times box is designed to perform this familiar multiplication process. It produces a hot output only when both inputs are hot. Finally, we need something that inverts sequences. Calling it a 'dash box,' this one takes any input sequence A and generates the output $\bar{A}$, value by value.

In this example the successive steps are $\bar{0} = 1$, $\bar{1} = 0$, and so on.

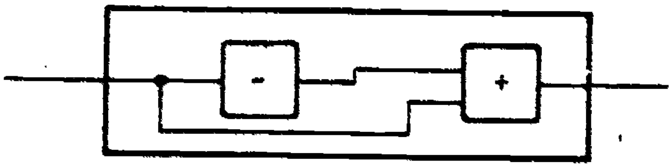
These +, × and - boxes provide a way of doing the computations of sequence arithmetic electrically. If you're still wondering what they look like inside, the answer is that modern devices are made of tubes or transistors, and you'll have to look elsewhere for the details. But it may be amusing for you to examine the following diagrams of old-fashioned +, × and - boxes and try to guess which is which. You'll probably need help even here, so don't forget that for our purposes it doesn't really matter.

PROBLEM 1. Identify these old-fashioned +, × and - boxes.



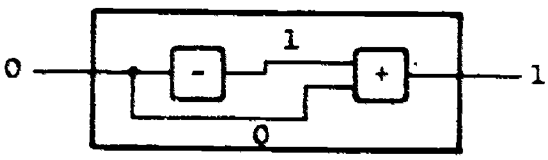
8.3 Electrical Machines: Analysis and Simplification.

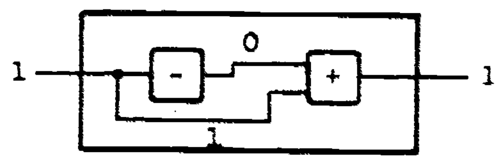
Suppose we connect +, × and - boxes together to make more complicated electrical machines. As a simple first illustration, let's figure out how Machine 1 behaves.



MACHINE 1

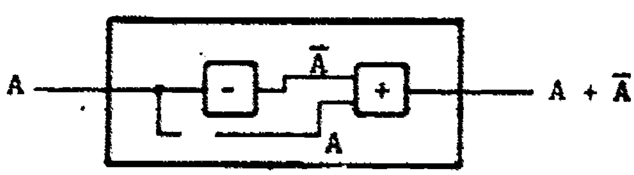
(Where wires are joined at a solid spot, as at the left of this diagram, it is to be understood that both are hot together or cold together. They carry identical sequences, or signals, and can be treated as a single wire.) Since there is only one input to the machine (at the left), we could simply ask what happens when that input is cold and what happens when it's hot. You can probably figure that out in your head, but these two diagrams may help.





Each wire is labeled, with 0 if it's cold and 1 if it's hot. In the first diagram the input is cold; in the second diagram the input is hot. But in both cases the output is hot! So whatever sequence of zeros and ones is fed into Machine 1, the output will always be hot. This is a brute force method of analyzing machine behavior, but it does work.

Now let's apply a more sophisticated analysis to the same machine. Whatever the input signal sequence is, call it A. Then the sequences, or signals, being carried in the various other wires can easily be labeled.

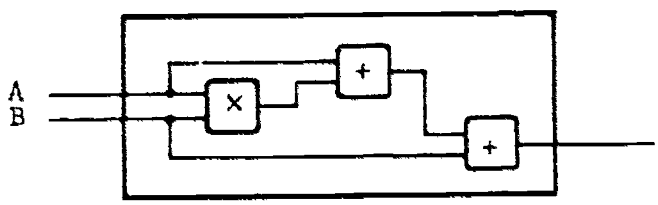


And the output is $A + \overline{A}$. But one of the theorems of sequence arithmetic guarantees

$$A + \overline{A} = I$$

so our output here is I, a sequence of ones. Brute force and strategy agree; the output of Machine 1 will always be hot, making it plain that the machine is actually useless. We can get an always hot output much more cheaply, just by connecting a wire to a wall outlet. The price of the dash and plus boxes can be saved.

Analyzing machines becomes more interesting when there are more inputs. For example, what does Machine 2 accomplish?



MACHINE 2

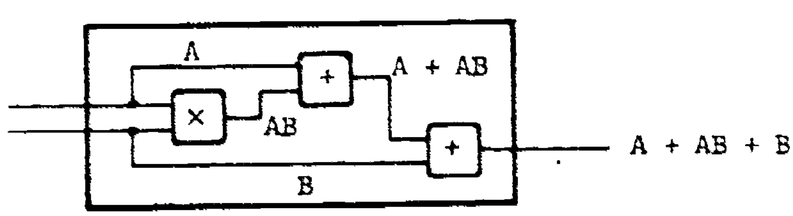
With just two inputs we could apply a brute force method, making one input the sequence

$$0\ 0\ 1\ 1$$

and the other input

$$0\ 1\ 0\ 1.$$

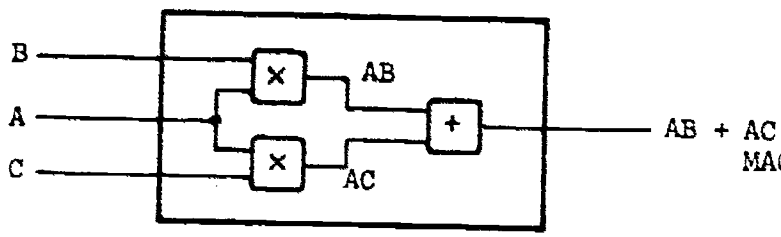
This would show us what happens in the only four possible situations this machine can face. But it's more elegant to use strategy. Labeling the inputs A and B, we proceed to identify the sequence, or signal, being carried in each wire. The results look like this.



The final output is $A + AB + B$. Can this be simplified? The combination $A + AB$ looks familiar, and checking our theorem list you would soon be reminded (see Theorem 16) that $A + AB = A$, for any pair of sequences A and B at all. The final output of Machine 2, namely, $A + AB + B$, can therefore be simplified to $A + B$, and the machine can be replaced by a single plus box.

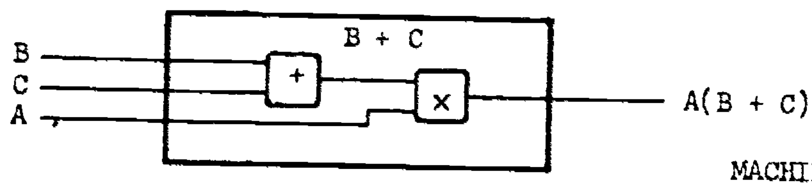
The other two boxes can be put back on the shelf for future use, or else returned for a refund.

As a third example, here is Machine 3. It has three inputs, A, B and C.



MACHINE 3

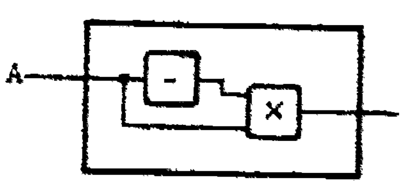
The sequences in all other wires have also been labeled, and the output is AB + AC. But Theorem 23 says that AB + AC is the same sequence as A(B + C), and so Machine 3 could be replaced by the simpler Machine 4.



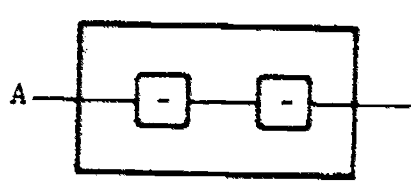
MACHINE 4

These three examples may be enough to suggest how the strategy of sequence arithmetic is used to analyze and simplify electrical machines.

PROBLEM 2. Label the sequence in each wire of Machines 5 and 6. Which of our theorems do they illustrate?

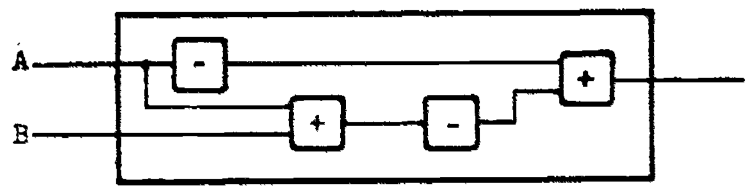


MACHINE 5



MACHINE 6

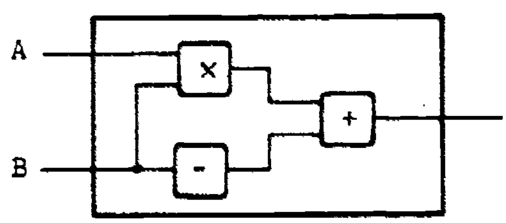
PROBLEM 3. Label the sequence in each wire of Machine 7.



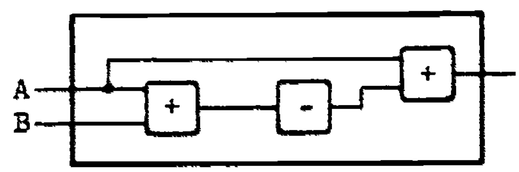
MACHINE 7

Show that the machine can be replaced by just one of our simplest boxes (+, × or -).

PROBLEM 4. Show that Machines 8 and 9 produce the same output sequence.

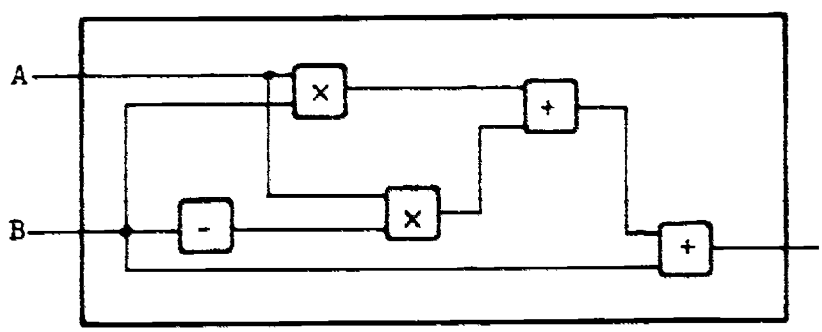


MACHINE 8



MACHINE 9

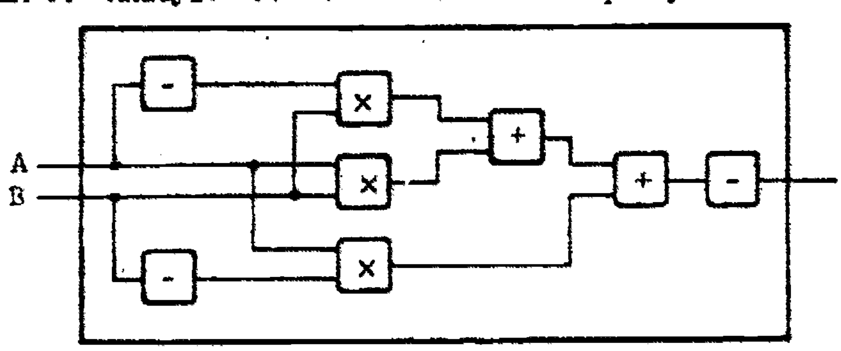
PROBLEM 5. Show that Machine 10 can be replaced by a single box (+, × or -).



MACHINE 10

(Where wires cross without a solid spot, as at the upper left of this diagram, there is no contact between them; they are insulated.)

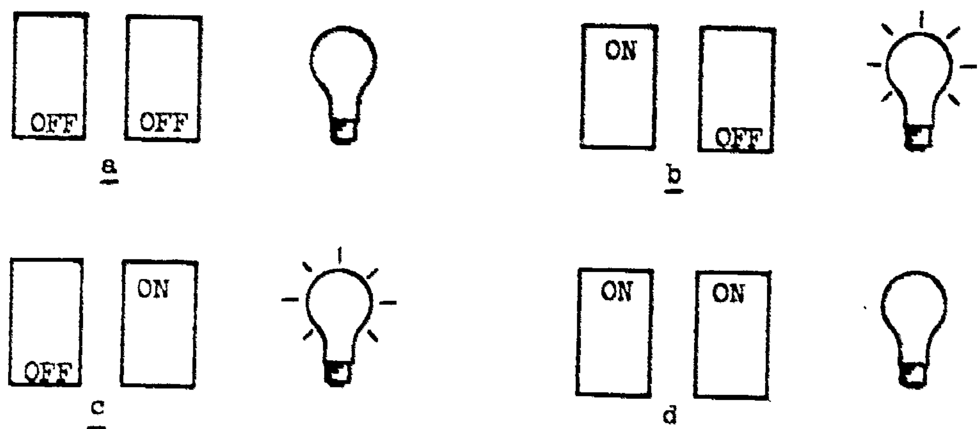
PROBLEM 6. Analyze Machine 11 and then simplify it to a two-box machine.



MACHINE 11

8.4 Electrical Machines: Design.

Let's turn now from the analysis of given machines to some examples of how machines can be designed for specific purposes. Once again we'll use light-hearted examples, but the spirit will be right. Take the problem of the hall light which is supposed to respond to either of two switches, one upstairs and one downstairs. If we suppose that the light should be off when both switches read off (see Diagram (a)) then it must come on when either one switch or the other is turned to the on position (Diagrams (b) and (c)) and it will go off again if both switches are turned on (Diagram (d)).



Think it over, and experiment with a real hall light if you have to, but the light should be on when the switches disagree and off when the switches agree. Using A and B to represent the two switches, and using 0 for off and 1 for on, what is required for the hall light is a sequence, or signal, which takes the value 1 whenever A and B disagree, and the value 0 otherwise. This can be summarized as follows.

Switch A:	0 0 1 1
Switch B:	0 1 0 1
Hall light:	0 1 1 0

Each column covers one of the only four situations that can occur. In the center two columns, for example, switches A and B disagree, so the hall light should be on. In the other two columns the switches agree, and so the light should be off. Only four columns appear because these represent the only four distinct situations which can arise. Now the problem is this. What combination of A and B behaves in this particular way? If you want to find out for yourself, then turn back to Chapter 4. In particular, notice once again how the special products AB, $A\bar{B}$, $\bar{A}B$ and $\bar{A}\,\bar{B}$ behave in the four distinct kinds of column we face. In particular, $A\bar{B}$ and $\bar{A}B$ behave like this.

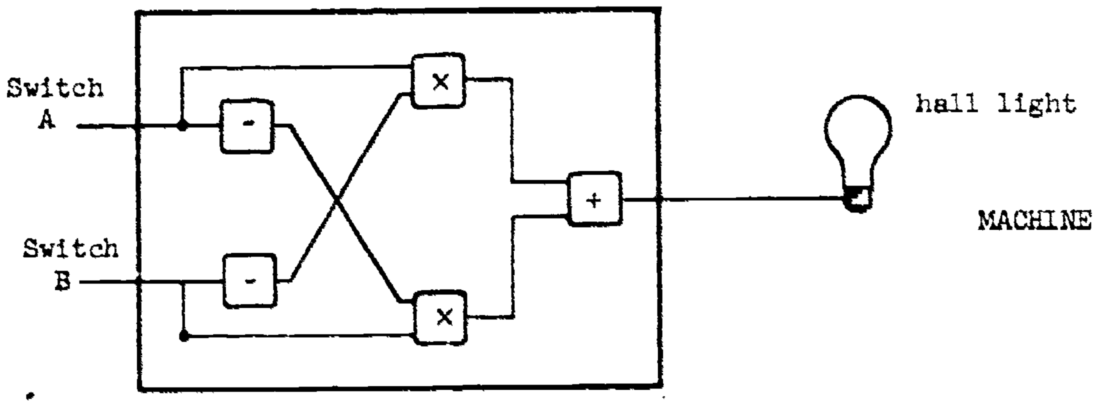
$A\bar{B}$: 0 0 1 0

$\bar{A}B$: 0 1 0 0

This suggests that a suitable sequence for the hall light will be the sum of these two products.

$$A\overline{B} + \overline{A}B: \quad 0\ 1\ 1\ 0$$

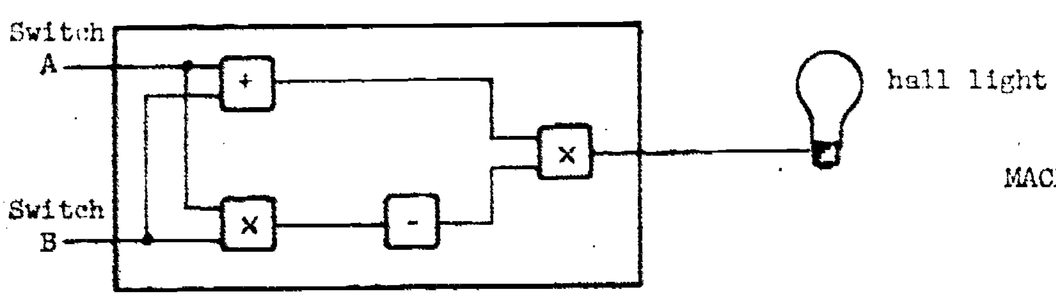
Now that we have a formula for the hall light sequence, it's an easy job to diagram the appropriate machine. (Machine 12.)



MACHINE 12

Label each wire yourself, according to the sequence it carries, and discover that the output is really $A\overline{B} + \overline{A}B$ so that this machine will properly control the hall light. Of course, there are much cheaper ways to provide proper control for a hall light, without using +, × and - boxes at all. But the point here is that we have designed an electrical machine for a specific task.

Checking back to Theorem 21 you will find that $(A + B)(\overline{A} + \overline{B})$ is the same as $A\overline{B} + \overline{A}B$. This suggests a second way in which a machine for hall light control could be designed. However, a count of the number of boxes required shows that in each case it takes <u>five</u> boxes to produce the output wanted, so there is no advantage in using the $(A + B)(\overline{A} + \overline{B})$ machine as a substitute. But looking at Theorem 22 does suggest a simplification. The sequence $(A+B)\overline{AB}$ is also the same as $A\overline{B} + \overline{A}B$ and it requires only four boxes. One dash box can be saved. Check the following diagram of Machine 13 which produces $(A + B)\overline{AB}$. Label each wire according to the sequence, or signal, which it carries.



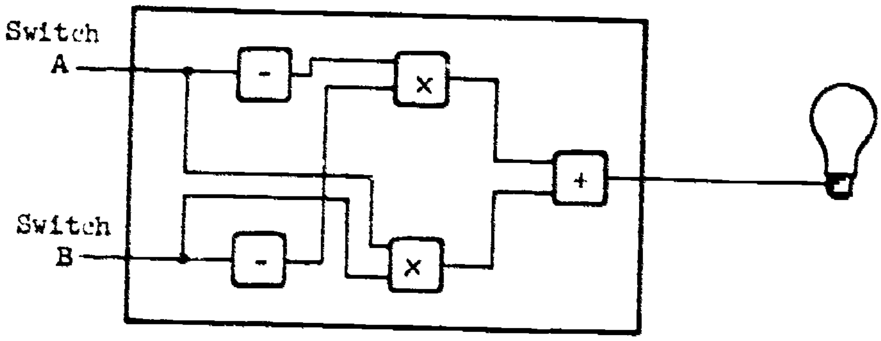
MACHINE 13

Machines which produce correct hall light behavior will find a completely different application in the next chapter.

Suppose we arrange that the hall light shall be <u>on</u> whenever the two switches A and B <u>agree</u>, and off otherwise. That's just the opposite of what we've just finished arranging, and it's not the approved system, but once in a while a hall light does get hooked up in this reversed fashion. The appropriate sequence for the light is now $AB + \overline{A}\,\overline{B}$ because this combination of A and B takes the value pattern we've asked for in the four familiar columns.

A:	0 0 1 1
B:	0 1 0 1
$AB + \overline{A}\,\overline{B}$:	1 0 0 1

Refer back to Chapter 4 again if you have to, but $AB + \overline{A}\,\overline{B}$ is what we need. And check the diagram of Machine 14, which produces $AB + \overline{A}\,\overline{B}$ as its output.



MACHINE 14

Perhaps you'll recall that $AB + \overline{A}\,\overline{B}$ is also the warrior's statement in the lady-tiger problem of Chapter 6. That problem and this modified hall light problem are the same mathematical problem. They differ only in the 'meaning' assigned to the various sequences (truth tables or electrical signals).

Next consider the problem of a hall light that has to respond to three switches A, B and C. If the light is off when all three switches read off, then it will have to go on when any one of the three switches is turned on. It will have to go off again if any two switches read on, but must come on again when all three switches read on. Think it over carefully, but the required behavior is summarized in this table where, as usual, 0 means off and 1 means on.

Switch A:	0 0 0 0 1 1 1 1
Switch B:	0 0 1 1 0 0 1 1
Switch C:	0 1 0 1 0 1 0 1
Hall light:	0 1 1 0 1 0 0 1

Can we design a machine which will output such a signal? If no inspiration strikes instantly, then there is always the method of products. To produce ones in columns 2, 3, 5 and 8 the basic products listed below will do. Verify this by referring back to the eight distinct columns of Chapter 4.

$\bar{A}\ \bar{B}\ C$	0 1 0 0 0 0 0 0
$\bar{A}\ B\ \bar{C}$	0 0 1 0 0 0 0 0
$A\ \bar{B}\ \bar{C}$	0 0 0 0 1 0 0 0
$A\ B\ C$	0 0 0 0 0 0 0 1

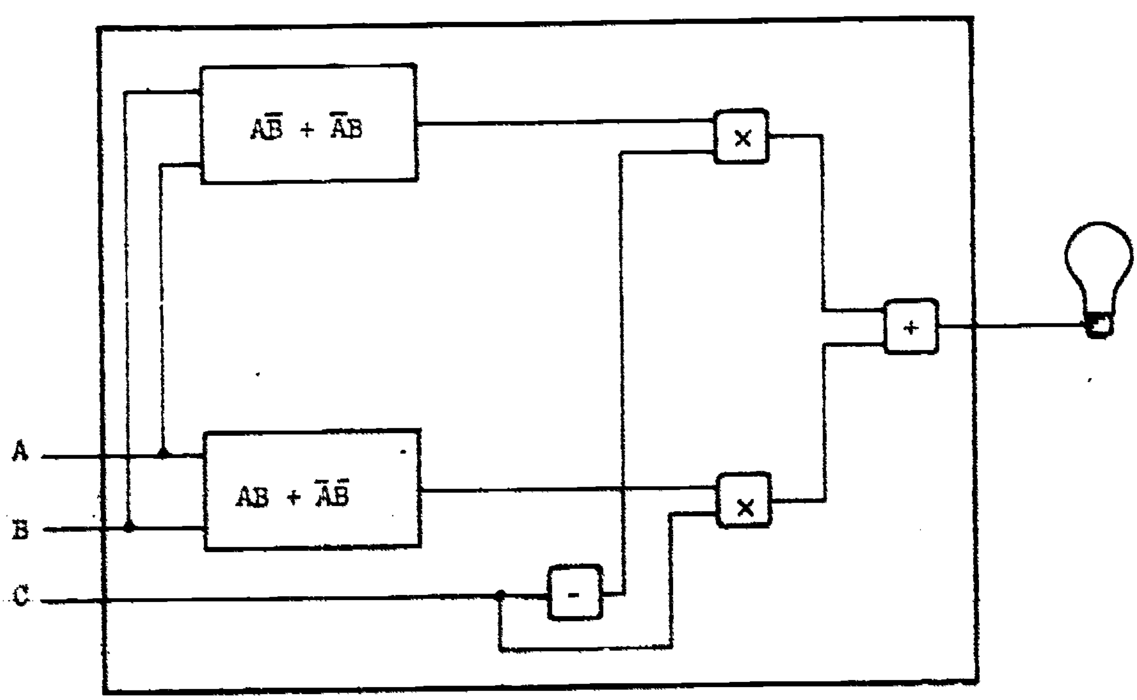
The hall light signal can, therefore, be achieved by adding these four products together.

$$\bar{A}\ \bar{B}\ C + \bar{A}\ B\ \bar{C} + A\ \bar{B}\ \bar{C} + A\ B\ C$$

But this would be a fairly expensive machine, requiring eight times boxes, three plus boxes and three dashes. It's only human to hope for simplifications. Unfortunately, very little simplification is possible in this example. We could regroup our four products into two pairs,

$$(AB + \bar{A}\ \bar{B})C + (A\bar{B} + \bar{A}B)\bar{C}$$

which eliminates two times boxes. This also allows us to use two machines we've designed before, for $AB + \bar{A}\ \bar{B}$ and for $A\bar{B} + \bar{A}B$. The following Machine 15 shows big boxes labeled $AB + \bar{A}\ \bar{B}$ and $A\bar{B} + \bar{A}B$. You know what's inside of them, or can easily look back to find out. As usual, check this machine to be sure you agree it's correct for the job.



MACHINE 15

PROBLEM 7. Design a machine which has this output pattern.

A: 0 0 1 1

B: 0 1 0 1

Output: 1 0 1 1

In words, the output is supposed to be hot except when A is cold and B is hot. One solution, using basic products, is

$$\text{Output} = \bar{A}\,\bar{B} + A\bar{B} + AB$$

but this uses more boxes than are necessary. Find a design which uses only two +, × or - boxes.

PROBLEM 8. Design a machine which has this output pattern.

A: 0 0 0 0 1 1 1 1

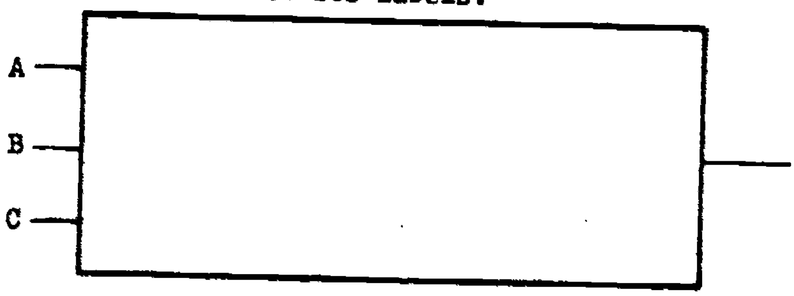
B: 0 0 1 1 0 0 1 1

C: 0 1 0 1 0 1 0 1

Output: 0 0 0 1 0 1 1 1

Notice that the output is hot only when the majority of A, B and C are hot. Four basic products will surely do the job here, but find a design which uses only five +, × or - boxes.

PROBLEM 9. Machine 16 has lost its labels.



MACHINE 16

To find out what it contains the brute force method of testing all possible input combinations is used. These familiar sequences include the eight possible combinations.

A: 0 0 0 0 1 1 1 1

B: 0 0 1 1 0 0 1 1

C: 0 1 0 1 0 1 0 1

The output sequence proves to be mostly hot.

Output: 0 1 1 1 1 1 1 0

From this evidence can you diagram the machine? (Hint: There are just six +, ×, - boxes involved.)

8.5 Summary.

So now 0 and 1 have <u>three</u> 'meanings,' the latest being cold for 0 and hot for 1. And sequences also have three meanings; as truth tables, as membership lists, and as electrical signals. And the mysterious $1 + 1 = 1$ has a third translation, as hot + hot = hot. The message of Chapters 6 to 8 has been that our abstract game seems to be perfectly modeled for applications of three quite different-looking types. Historically it is the applications which came first, developing more or less independently of each other. Later the deep analogy between them was gradually recognized and the abstract game began to develop. Each application then served to help the others and strategy grew quickly, another example of 'in union there is strength.' Progress from applications to abstraction is characteristic of mathematics.

Chapter 9

DESIGNING A COMPUTER

9.1 Binary Symbols for Numbers.

Our story now takes a surprising turn, in the direction of 'ordinary' arithmetic. Ordinary arithmetic is a much better-known game than sequence arithmetic, and a more complicated one. The pieces with which it is played are called numbers and, like any other game, it has its own basic rules and strategy. Starting from such simple and traditional origins as

$$1 + 1 = 2$$

the game progresses ultimately to the sophisticated heights of calculus, and beyond. It will be reassuring to hear, however, that only the basement level of that towering skyscraper of theory will be involved here.

The first things we will need are called the binary symbols for numbers. Decimal symbols are, of course, more popular, at least with human beings. Assigning the symbols 0 to 9 to the simplest numbers, the decimal system then uses the idea of position value to build symbols for more complicated numbers. As everyone knows perfectly well,

$$2468$$

represents the combination

$$2 \times 1000 + 4 \times 100 + 6 \times 10 + 8 \times 1.$$

The key symbols in this system are 1, 10, 100, 1000, and so on, the value of the digit 1 increasing tenfold with each shift to the left. The binary system is very similar. It uses only the digits 0 and 1, but position value remains the central idea. Here are a few of the key binary symbols, accompanied by their decimal translations. Notice that the value of the digit 1 now increases twofold with each shift to the left.

Binary	Decimal
0	0
1	1
10	2
100	4
1000	8
10000	16
100000	32

Just as with decimals, the binary symbols for other numbers can be fashioned by choosing suitable combinations.

Binary	Decimal
1 1	3 = 2 + 1
1 0 1	5 = 4 + 1
1 0 1 1	11 = 8 + 2 + 1
1 1 0 0 1 0	50 = 32 + 16 + 2

In the last example the three 1's translate to 32, 16 and 2, and 1 1 0 0 1 0 is an alias for the number we call decimal 50.

The idea of binary symbols is basically a simple idea. It can be extended with very little difficulty to the other numbers of ordinary arithmetic. For instance, a minus sign still denotes a negative number, so - 1 0 is binary for -2. And

.1

represents the same number as $\frac{1}{2}$, while

.01

is an alias for $\frac{1}{4}$. Since we will be using integers only, there is no immediate need for detailed exploration to the right of the 'binary point.'

PROBLEM 1. Translate from decimal to binary.

Binary	Decimal	Binary	Decimal
	6 = 4 + 2		15
	7 = 4 + 2 + 1		17
	9		28
	10		59

PROBLEM 2. Translate from binary to decimal.

Binary	Decimal	Binary	Decimal
1 0 1 1		1 1 1 1 1	
1 1 0 0		1 0 0 1 0 0	
1 0 1 0 0		1 0 1 1 0 1	
1 1 0 0 1		1 1 1 1 1 1	

PROBLEM 3. Suppose three subsets A, B and C have these membership lists. (The top row gives each member a number, from 0 to 7.)

	0	1	2	3	4	5	6	7
A:	0	0	0	0	1	1	1	1
B:	0	0	1	1	0	0	1	1
C:	0	1	0	1	0	1	0	1

It's easy to discover that this puts one member into each of the eight basic products $\bar{A}\,\bar{B}\,\bar{C}$, $\bar{A}\,\bar{B}\,C$, $\bar{A}\,B\,\bar{C}$, and so on. It's also easy to see that each member's number is duplicated <u>in binary</u> in the column underneath his number. (Under 3, for example, you find 0 1 1.) Write the number of each member in the part of our standard subset diagram where he belongs. (Members 3 and 4 have already been placed.)

	3
4	

This explains the numbering pattern used in Chapter 7. You may want to extend the pattern to two-subset or four-subset diagrams.

9.2 <u>Binary Computing</u>.

Next let's notice how easily the sums and products of ordinary arithmetic can be computed using binary symbols. For sums, the four basic facts are these.

$$\begin{array}{r}0\\+\,0\\\hline 0\end{array}\qquad\begin{array}{r}0\\+\,1\\\hline 1\end{array}\qquad\begin{array}{r}1\\+\,0\\\hline 1\end{array}\qquad\begin{array}{r}1\\+\,1\\\hline 1\,0\end{array}$$

The last of these is our old, familiar $1 + 1 = 2$ in binary translation, and it shows that whenever we face the sum $1 + 1$, we are going to have to 'put down 0 and <u>carry</u> 1,' just as with decimal symbols the sum $5 + 7 = 12$ makes us 'put down 2 and carry 1.' The technique will look very familiar to you. Here are a few illustrations.

$$\begin{array}{r}1\,0\,1\,0\\1\,0\,0\\\hline 1\,1\,1\,0\end{array}\qquad\begin{array}{r}1\,0\,1\,0\,1\\1\,0\,0\\\hline 1\,1\,0\,0\,1\end{array}\qquad\begin{array}{r}1\,0\,1\\1\,1\,1\\\hline 1\,1\,0\,0\end{array}$$

In the first illustration there are no carries at all. In the second there is just one, and it amounts to a binary translation of $4 + 4 = 8$. The third involves several carries which can be similarly explained. You can see that binary adding is a simple enough process.

A somewhat different view of this matter of carries rates at least passing mention, because it exposes new mathematical horizons. Whenever the sum $1 + 1$ appears in a column, then as far as that particular column is concerned <u>the sum is 0</u>. Of course, we know that $1 + 1$ is really 10, or 2 if you prefer, at least in ordinary arithmetic. But <u>the two is thrown away</u>, into the next column to the left. It becomes a carry. In decimal computations tens are thrown away, into the next column to the left. The idea of throw-aways has turned out to be surprisingly useful, with the result that the inevitable process of abstraction has run its course. 'Throw-away arithmetics' are now official parts of the collection of games that we call mathematics. In particular, 'throw-away twos' is played with only two pieces, 0 and 1. Its basic rules include no surprises except for

$$1 + 1 = 0$$

from which it gets its name. As you'll see shortly, binary computations are very popular with electrical calculating machines, so that 'throw-away twos' sees heavy, but more or less out-of-sight, action every time that a carry is made. Throw-away arithmetics also have other more exotic and less obscured applications, to the production of random numbers for example, and to the design of experimental patterns. But that's another story so let's get back to our sequences of zeros and ones.

Here is one self-explanatory example of multiplication using binary symbols.

$$\begin{array}{r}
1\,0\,1\,1 \\
\underline{1\,0\,1} \\
1\,0\,1\,1 \\
0\,0\,0\,0\;\; \\
\underline{1\,0\,1\,1\;\;\;\;} \\
1\,1\,0\,1\,1\,1
\end{array}$$

In decimal this would read $11 \times 5 = 55$.

The other operations of ordinary arithmetic can also be performed with binary symbols, but we won't go into the details here.

PROBLEM 4. Perform these additions. Also translate to decimal.

$$\begin{array}{r} 1\,0\,1\,0 \\ \underline{1\,0\,1\,1} \end{array} \qquad \begin{array}{r} 1\,1\,0 \\ \underline{1\,0\,1} \end{array} \qquad \begin{array}{r} 1\,1 \\ \underline{1\,1} \end{array} \qquad \begin{array}{r} 1\,0\,1\,0\,1 \\ \underline{1\,0\,1\,1\,1} \end{array}$$

PROBLEM 5. Multiply 1 1 0 by 1 0 1. Also translate to decimal.

9.3 The Half-Adder.

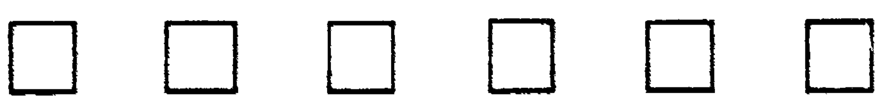
Now we come to the reasons for this excursion into binary symbolism. All these 0's and 1's must be at least slightly reminiscent of sequence arithmetic. A binary symbol such as

1 0 1 1 0 1

is a sequence of zeros and ones. It's true that in this chapter you've been asked to interpret this sequence in another new way, but it's still a sequence of zeros and ones. With punched cards, of the sort mentioned in Chapter 8, we could even convert this sequence into electricity,

h e h h e h

and our old friend the number 45 (decimal) would take a form that an electrical machine can understand. That's our first point. Binary symbols can be used to make the numbers of ordinary arithmetic understandable to electrical machines. Which of these six cards should be punched to translate 45 into electricity?



With numbers represented in electrical form it isn't too big a jump to the design of an electrical machine that will do computations. Take the operation of addition. What would a machine have to be able to do in order to compute sums as we did just a few moments ago? Principally it would have to know how to handle the four basic sums,

$$\begin{array}{r} 0 \\ \underline{+\,0} \end{array} \qquad \begin{array}{r} 0 \\ \underline{+\,1} \end{array} \qquad \begin{array}{r} 1 \\ \underline{+\,0} \end{array} \qquad \begin{array}{r} 1 \\ \underline{+\,1} \end{array}$$

because computations involve repeated handling of these four. In each case it must know what to 'put down' and what to 'carry.' The facts are simple enough to be summarized in the four columns of this little table.

A:	0	0	1	1
B:	0	1	0	1
Record:	0	1	1	0
Carry:	0	0	0	1

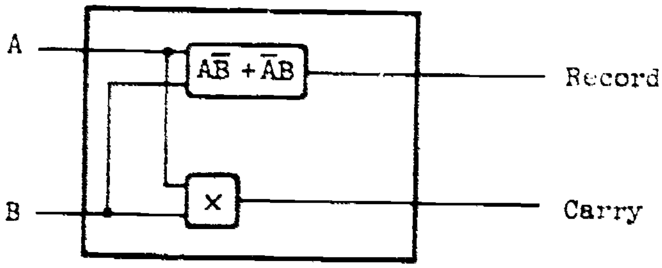
The two digits to be added are called A and B. For added dignity, the table uses the word 'record' instead of 'put down.' Examine these columns to be sure you agree that they properly present the facts of addition. Then let's concentrate on the row labeled 'record.' When should a 1 be recorded? Only when the two digits to be added disagree! That may hit a resonant spot in your memory. Our four columns here are the same four distinct kinds of column that we've encountered before. And record takes the value 1 only in the two columns where A and B disagree. Do you recall that such behavior arises in sequence arithmetic when

$$A\bar{B} + \bar{A}B$$

is computed? And for computing such a sequence we have already designed (in Chapter 8) at least two electrical machines. Any one of those machines can now be used to produce the proper record. We simply have to offer it A and B as inputs. But we also need a carry, so look at the bottom row of the above table. The only 1 is in column four, where A and B are both 1. This behavior may recall the product

$$AB$$

of sequence arithmetic. So a simple × box will produce the correct carry as output, given A and B as inputs. To obtain both of the outputs we need, the following machine will serve. It is called a half-adder.



HALF-ADDER

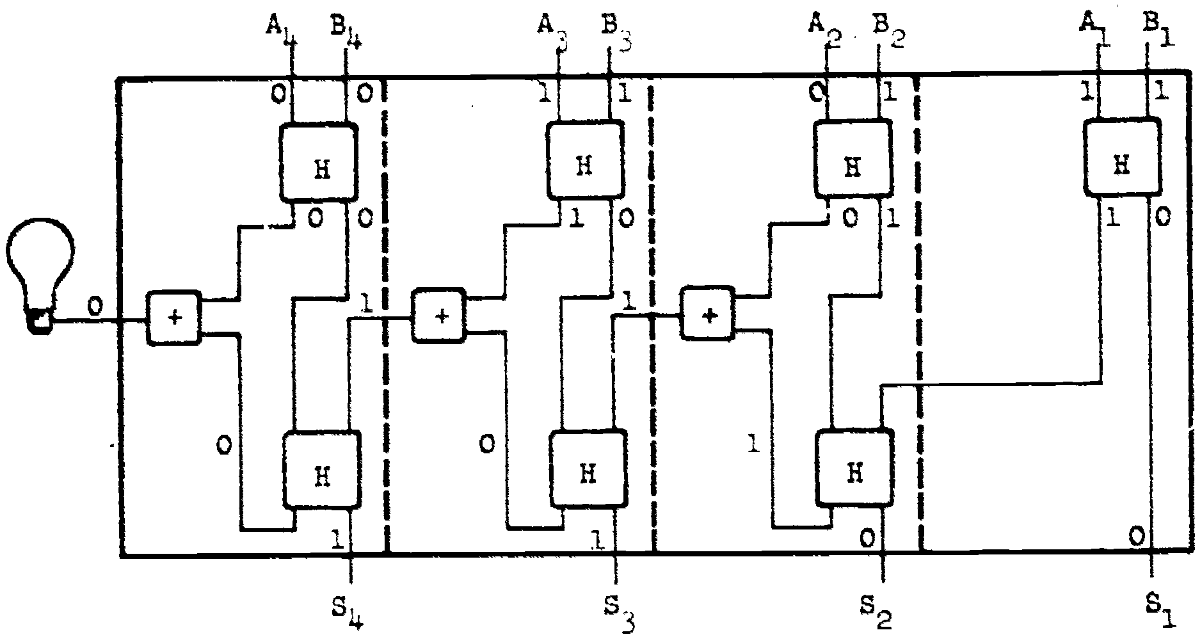
This machine can handle only two digits. It will take a combination of half-adders to compute an ordinary sum.

9.4 An Adding Machine.

Suppose these four-digit numbers are to be added. Each A or B is either a 0 or a 1.

$$\begin{matrix} A_4 & A_3 & A_2 & A_1 \\ B_4 & B_3 & B_2 & B_1 \end{matrix}$$

Let me try to convince you that the following machine is up to the job. It will be easier than a first glance might suggest.



The letter H represents a half-adder.

The best way to understand this adding machine is to actually follow it through a typical computation. Take these numbers.

$$A_4 \ A_3 \ A_2 \ A_1 = 0\ 1\ 0\ 1$$

$$B_4 \ B_3 \ B_2 \ B_1 = 0\ 1\ 1\ 1$$

A few moments ago we added them and got 1 1 0 0, confirming the fact that 5 + 7 is still 12. For the machine to duplicate our effort these A's and B's must be brought to the eight input contacts. At the top of the diagram you can see 1's and 0's (for hot and cold) labeling the appropriate wires. Other 1's and 0's indicate the repercussions within the machine. Follow the action slowly from right to left, just as though you were doing the computation by hand. With record coming out of the lower right of each H box, and carry coming out of the lower left, you will find that the machine takes exactly the same steps that you would take, and hopefully arrives at the same result. That result appears at the four output contacts, at the bottom of the diagram.

$$S_4 \ S_3 \ S_2 \ S_1 = 1\ 1\ 0\ 0$$

This machine was designed to handle binary symbols of four digits each. Notice the dotted lines which separate it into four parts. Each of these parts (except the one at the right) is called a <u>full-adder</u>. A full-adder takes the A and B digits of one column and the carry from the previous column (to the

right) as inputs; it outputs one correct digit for the sum and the carry for the next column (to the left). For serious computing a machine must handle symbols of roughly forty binary digits (bits for short). To do this, more full-adders will have to be added at the left of our diagram, until the selected capacity is reached. Whatever capacity is selected, it is, of course, possible to offer the machine numbers whose sum will exceed that capacity. In our simple four digit machine any sum over fifteen will do just that. Such a sum will produce a hot carry out of the leftmost + box, causing the 'overflow' light to come on as a warning that a most important digit, the one with the highest position value, has just escaped from the computation. As an example, follow the computation

$$\begin{array}{r} 1\ 1\ 0\ 0 \\ +\ 0\ 1\ 0\ 0 \end{array}$$

through the machine. The overflow light should go on. What sum does our machine produce?

9.5 Computer Science.

One method of computing sums electrically has just been outlined. There are many alternative methods. It is also possible to design machines which perform the other operations of ordinary arithmetic, and machines which perform various related chores which will be described in our next and final chapter. By connecting these various machines together, a remarkably versatile device can be constructed, capable of doing almost anything arithmetical, and at electrical speed. The literature of computing machine design carries the full story and can be studied by embryo computer scientists. The main point of this chapter has been that binary symbols offer a way by which numbers can be represented electrically, as sequences of zeros and ones, and that sequence arithmetic, in which

$$1 + 1 = 1$$

plays a basic role in the design of electrical machines which do correct computations for a different arithmetic in which

$$1 + 1 = 2.$$

PROBLEM 6. Let A, B and C denote the three inputs to a full-adder, C being the carry from the previous column. The usual three sequences

A: 0 0 0 0 1 1 1 1

B: 0 0 1 1 0 0 1 1

C: 0 1 0 1 0 1 0 1

display the only eight possible combinations a full-adder can ever face, one combination in each column. What should the outputs of the full-adder be for each of these combinations?

Record:

Carry:

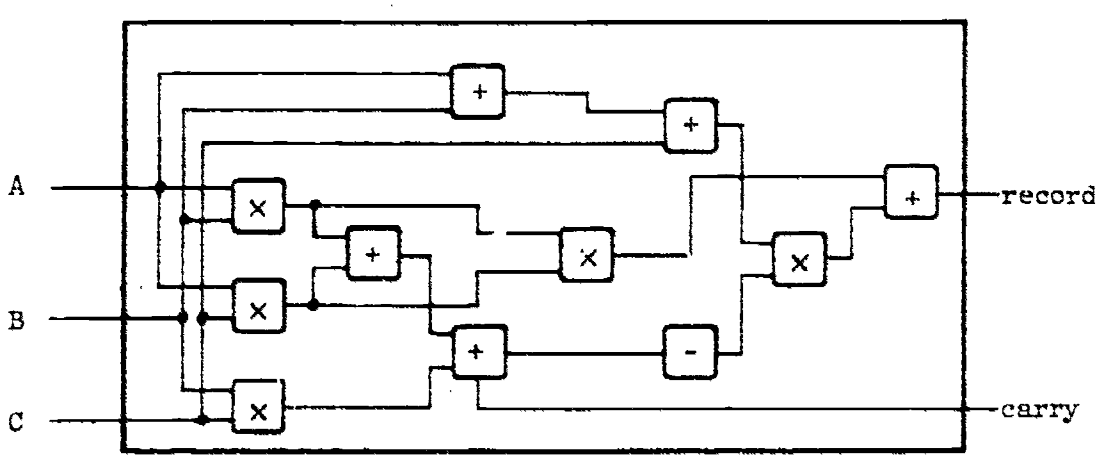
Convince yourself that the following symbols of sequence arithmetic properly represent record and carry.

$$\text{Record} = A\,B\,C + A\,\overline{B}\,\overline{C} + \overline{A}\,B\,\overline{C} + \overline{A}\,\overline{B}\,C$$

$$\text{Carry} = AB + BC + CA$$

Design an alternate full-adder from the above two symbols. Does it appear to be simpler than the design given in Section 9.4, or not?

PROBLEM 7. Use our theorems to convince yourself that the following design also represents a correct full-adder.



PROBLEM 8. Show that what the full-adder described in Section 9.4 actually computes are

$$\text{Record} = [(A + B)\overline{AB} + C]\overline{(A + B)\overline{AB}C}$$

$$\text{Carry} = AB + (A + B)\overline{AB}C$$

and use our theorems to verify that these are aliases for the symbols of Problem 6.

Chapter 10

THE COMPUTER IN ACTION

10.1 Memory.

The previous chapter has suggested that it is possible to design an electrical machine which can perform the operations of ordinary arithmetic. Not only is it possible but, as most of you know, thousands of such machines already inhabit our planet. The computing abilities of such machines would be largely wasted if they were unable to 'remember' the numbers involved, and so, many ways have been devised to provide machines with memories. All that is needed is a way to preserve electrical sequences, so that they can be retrieved when wanted. One of the easier devices to picture in your mind consists of rows of magnetizable spots. This little memory, for example (black spots magnetized, white spots not),

○ ○ ●
○ ● ○
● ○ ○
● ● ●

can be translated for human computers into

0 0 1
0 1 0
1 0 0
1 1 1

and, if you want to, you can find a connection between this miniature memory and the three-way hall light of Chapter 8. A human computer uses his brain, assorted sheets of paper, and perhaps still other apparatus in various stages of disorganization, for memory. In a machine, however, the sequences of zeros and ones are neatly stored in rows of uniform length, one sequence to a row. Each row is called a memory location and numbered for easy reference, using binary symbols. The four locations above could be labeled 0 0 1, 0 1 0, 0 1 1 and 1 0 0. A larger memory appears below. It will serve as the conversation piece of this chapter. The memory itself consists of the column of nine-value sequences in the center. Location numbers, in binary and decimal, have been included at the left, for your convenience as we refer back to this memory. Some explanatory phrases are also offered at the right, but these will not be

clear until later. This collection of ones and zeros will look formidable at first sight, but it is a tiny memory by the standards of serious modern computing.

	Location	Sequence	Explanation
1	000001	001010000	Add an X number to SUM.
2	000010	010010011	
3	000011	011010000	
4	000100	001010001	Add 1 to COUNT.
5	000101	010010010	
6	000110	011010001	
7	000111	111001111	Is COUNT less than 32 ?
8	001000	110001011	
9	001001	100010000	<u>No</u>. Punch SUM.
10	001010	000000000	Stop.
11	001011	0●1000010	<u>Yes</u>. Modify instruction number 2.
12	001100	010010010	
13	001101	011000010	
14	001110	101000001	Jump back to 1.
15	001111	000100000	THIRTY-TWO
16	010000	000000000	SUM
17	010001	000000000	COUNT
18	010010	000000001	ONE
19	010011	011001111	X_1
20	010100	111101001	X_2
21	010101	011000011	X_3
22	010110	101001110	X_4

(Locations 23 to 50 are filled with other sequences that will be called simply X_5, X_6, up to X_{32}.)

10.2 <u>Instructions</u>.

If machines are to perform arithmetical tasks for human masters, then communications between man and machine must be established. A language understandable by both must be devised. The machine must be told what to do. It needs <u>instructions</u>. As a matter of fact, some of the sequences in the above memory <u>are</u> instructions. It is only necessary that we and the machine both

understand the language. These instructions can be explained to you in moderately plain English, but they will have to be explained to the machine by proper electrical wiring. In previous chapters we've designed machines for adding numbers as well as for various simpler tasks. Here we will need similar machines for understanding and executing eight types of instruction, but we'll leave the details of design in the capable hands of computer scientists and focus our attention on a strictly human to human understanding. Taking last things first, one type of instruction is

0 0 0 ______	STOP

the last six digits of the sequence being irrelevant. As indicated, 0 0 0 tells the computer to stop. A more typical kind of instruction is

0 0 1 ______	Copy the sequence which is now in location ______ into location 0 0 0 0 0 0, erasing the previous content of 0 0 0 0 0 0 first. The sequence originally in ______ should be in both locations after this instruction is executed.

The English translation is at the right. Notice that of the nine digits in the instruction, the first three indicate what kind of job is to be done (0 0 1 means a copying job) and the last six indicate a memory location which is involved. Location 0 0 0 0 0 0 is a special memory location which sees a great deal of action once the computation gets underway. This will all be much clearer when we have followed the maching in action for a while, as we will shortly, but let's just list the other instruction types first.

0 1 0 ______	Add number in location ______ to number in 0 0 0 0 0 0, leaving content of ______ unchanged. The sum should appear in 0 0 0 0 0 0. This addition is a binary computation, using full-adders as described in Chapter 9, and not the simple 1 + 1 = 1 addition of Chapter 2.
0 1 1 ______	Copy sequence now in location 0 0 0 0 0 0 into location ______, erasing any previous content of ______. The sequence originally

in 0 0 0 0 0 0 should be in both locations after this instruction is executed.

1 0 0 _ _ _ _ _ _	Punch the sequence in _ _ _ _ _ _ onto a card. (The sequence also remains in location _ _ _ _ _ _ for future use.)
1 0 1 _ _ _ _ _ _	Take the next instruction from location _ _ _ _ _ _. (This breaks the normal routine, in which instructions are taken from consecutive locations. It is called a <u>jump</u> instruction.)
1 1 0 _ _ _ _ _ _	Take the next instruction from location _ _ _ _ _ _ <u>if</u> the sequence now in location 0 0 0 0 0 0 is negative. Otherwise follow the normal, consecutive routine. (This requires the machine to make a decision.)
1 1 1 _ _ _ _ _ _	Subtract the sequence in _ _ _ _ _ _ from the sequence in 0 0 0 0 0 0, leaving content of _ _ _ _ _ _ unchanged. The difference should appear in 0 0 0 0 0 0.

Now let's watch the action as these types of instruction are executed.

10.3 A Program.

The sequences in the memory exhibited in Section 10.1 give the computer full instructions for performing a particular arithmetical job. Such a set of instructions is called a <u>program</u>. Let's see what job this program spells out. The first instruction is in location 1, the next in location 2, and so on. See if you agree that this is what happens when the first three instructions are executed.

1. 0 0 0 0 0 0 0 0 0 appears in location 0.
2. 0 1 1 0 0 1 1 1 1 appears in location 0.
3. 0 1 1 0 0 1 1 1 1 appears in location 16.

The sequences in locations 16 and 19 have been added, and the sum (X_1) has been stored in location 16. Try the next three instructions in our computer memory. Here is what they achieve.

4. 0 0 0 0 0 0 0 0 0 appears in location 0.
5. 0 0 0 0 0 0 0 0 1 appears in location 0.
6. 0 0 0 0 0 0 0 0 1 appears in location 17.

The computer has 'counted' from 0 to 1 in location 17. Take the next two instructions together. When a sequence stands for a number instead of for an instruction, the first digit is used to indicate the sign, 0 meaning a positive number and 1 a negative number. The other eight digits are the binary symbol for the number itself. As instruction 7 comes up the sequence 0 0 0 0 0 0 0 0 1 is still in location 0. Subtracting 32 will produce a negative 31.

7. 1 0 0 0 1 1 1 1 1 appears in location 0.
8. The computer 'jumps' to location eleven.

Instructions 9 and 10 are bypassed for the moment. Now comes a crucial development. Take three more instructions together.

11. 0 1 0 0 1 0 0 1 1 appears in location 0.
12. 0 1 0 0 1 0 1 0 0 appears in location 0.
13. 0 1 0 0 1 0 1 0 0 appears in location 2.

The content of location 2 has been modified. In a moment the computer will be executing instruction 2 again, and you should note the effect of this modification. We're up to instruction 14

14. The computer 'jumps' to location 1.

We've watched one trip through what is called a 'loop.' Let's watch one more trip.

1. 0 1 1 0 0 1 1 1 1 appears in location 0.
2. 1 0 0 0 1 1 0 1 0 appears in location 0
3. 1 0 0 0 1 1 0 1 0 appears in location 16.

The sequences in locations 16 and 20 have been added, and the sum $(X_1 + X_2)$ has been stored in location 16.

4. 0 0 0 0 0 0 0 0 1 appears in location 0.
5. 0 0 0 0 0 0 0 1 0 appears in location 0.
6. 0 0 0 0 0 0 0 1 0 appears in location 17.

The computer has counted from 1 to 2 in location 17.

7. 1 0 0 0 1 1 1 1 0 appears in location 0.

8. The computer jumps to location eleven.

11. 0 1 0 0 1 0 1 0 0 appears in location 0.

12. 0 1 0 0 1 0 1 0 1 appears in location 0.

13. 0 1 0 0 1 0 1 0 1 appears in location 2.

The instruction in location 2 has again been modified.

14. The computer jumps to location 1.

And two trips through the 'loop' have been completed. Further trips I leave to you to follow in detail. The third trip will first bring the sum $X_1 + X_2 + X_3$ into location 16. Then it will raise the count in location 17 to three. Finally, it will modify the instruction in location 2 to read 0 1 0 0 1 0 1 1 0 Convince yourself of these things and then convince yourself that the sum in location 16 will continue to develop until it includes whatever numbers have been stored in locations 19 through 50, at which point it will be $X_1 + X_2 + \ldots + X_{32}$. Thirty-two numbers will have been summed. An important change will then occur. The computation will break out of the loop! Do you see how that happens? On the thirty-second trip through the loop, after all thirty-two numbers have been summed, the count (in location 17) will climb to 32. Then notice the result of executing instructions 7 and 8.

7. 0 0 0 0 0 0 0 0 0 appears in location 0.

8. The computer decides not to jump.

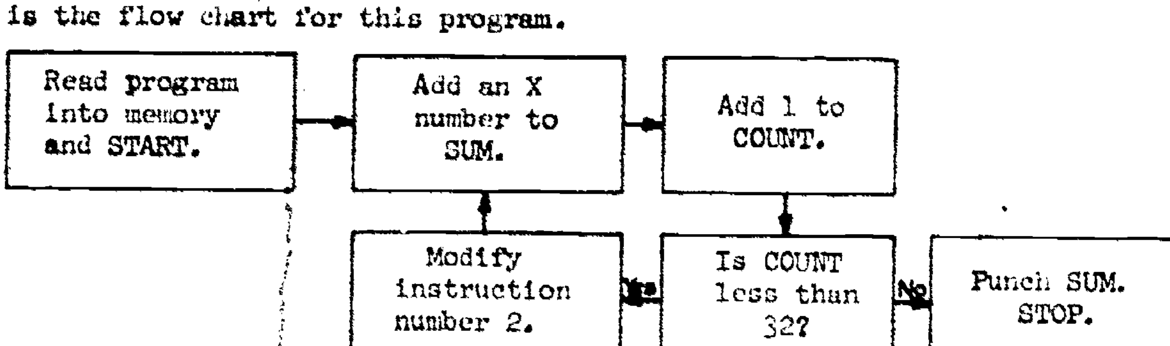
For the first time it refuses the jump, because now the number in location 0 is not negative. Instructions 9 and 10, which have been bypassed thirty-one times, finally get their turn.

9. The computer punches out the final sum.

10. The computer stops.

Its assigned job has been completed. As you can now see, that job was the summing of thirty-two numbers.

To summarize the action, a flow chart is a helpful and common device. Here is the flow chart for this program.



The loop is clearly visible. After thirty-one complete trips around this loop, the 'No' exit is taken and the loop has been broken. It takes a modest effort to see clearly all the details of this electrical computation. But once you've mastered those details, the precise logic of the program has an almost aesthetic appeal.

Two final remarks may be useful. Plainly a sequence may be either a number or an instruction. How effectively this fact is exploited can be seen by recalling the modifications made to instruction 2. By treating this sequence as a <u>number</u>, and adding one to it, the machine converts the sequence into the desired <u>instruction</u> for the next trip around the loop. This versatility can also lead to catastrophe, as you will soon see. The second remark is this. The program could easily be modified to sum a million or more numbers instead of just thirty-two. Our present machine, however, has its limitations. Memory locations 51 to 63 were not used, so with a few changes thirteen more numbers could have been accommodated. Thirty-two is such a pleasant number in binary that this extra capacity was ignored in my example, but see Problem 13.

10.4 <u>A Post-Mortem.</u>

It is easy to see that the tiniest program error, a single 0 where there should be a 1 or vice versa, can produce a catastrophe. As an easy example, what would happen if the sequence for location 1 were mispunched and entered the computer as 0 0 0 0 1 0 0 0 0, the error being in the third digit? You probably see at once that this makes the very first instruction a stop instruction. The computation will never even get started. When the start button is pushed, it may appear that the computer has not been plugged in. But experience has shown that when a program fails to run properly it is usually the program and not the machine which is to blame. We just investigated the effect of a known error in a known location. But hunting down a program error is usually hard work and ways have been found for the computer to assist in the search. Our program is so brief that in the event of trouble we could ask the machine to punch out the entire memory for our inspection. In a more serious problem such a 'memory dump' would be too voluminous to be helpful, and more sophisticated detective work is called for. Here is a typical, but simplified, example.

Suppose we know that our program should take less than a minute of the machine's time. It has been running, however, for three minutes and the operator has just stopped the computation. A program error is suspected, but in which memory location? To find the error a 'post-mortem' is conducted, in

which the machine punches out the answers to our questions. Here is a plain English translation of the process.

> Question: Which was the last instruction executed?
>
> Answer : Instruction in location 22.

This is surely a discouraging reply. The sequence in 22 was supposed to be our <u>number</u> X_4. The machine has interpreted it as an <u>instruction</u>. Some numbers also make perfectly reasonable instructions and the sequence in location 22

1 0 1 0 0 1 1 1 0

means 'Jump to location 14 for your next instruction' just as surely as it means 'negative 78.' So the machine has jumped to location 14. Let's follow it.

> Question: What is the sequence in location 14 ?
>
> Answer: 1 0 1 0 0 0 0 0 1

Just what it should be, a 'Jump to location 1' instruction.

> Question: What is the sequence in location 1 ?
>
> Answer: 0 0 1 0 1 0 0 0 0

Just what it should be, 'Copy SUM from location 16.' So we follow the machine to location 2.

> Question: What is the sequence in location 2 ?
>
> Answer: 1 0 1 0 1 0 1 1 0

Ouch! This doesn't remotely resemble the 'add' instruction we thought we had here. Instead it reads 'Jump to location 22 for your next instruction.' And now we know why the machine wouldn't stop. It has been patiently following instructions, jumping from 22 to 14 to 1 to 2 to 22 over and over again, following an unintended loop. The program error, which we still have not located, has sent the machine into a senseless, never-ending loop, in which it computed nonsense until the operator mercifully stopped it. Such futile and unending loops have been compared with human insanity. Our machine was unable to help itself, and required shock treatment (cutting off the electric power).

Now we know <u>what</u> happened to our program. But <u>why</u>? Either our second instruction was incorrect when it entered the computer, or it was spoiled afterward. Let's try the second possibility first.

> Question: What sequences are in locations 11, 12 and 13 ?
>
> Answer: 0 0 1 0 0 0 0 1 0
> 0 1 0 0 1 0 1 0 0
> 0 1 1 0 0 0 0 1 0

This looks like the error! The middle sequence has a displaced 1. It should read 0 1 0 0 1 0 0 1 0 instead. The effect of this misplaced 1 is catastrophe. This twelfth instruction was supposed to alter instruction 2 by adding to it the sequence in location 18. Instead, it adds the sequence in location 20, and alters instruction 2 to the jump instruction

1 0 ' 0 1 0 1 1 0

that we discovered a few moments . Check for yourself that the misplaced 1 in instruction 12 fully explains what happened to our program. Follow the computation until it enters the senseless, unending loop. If in our post-mortem we had asked for the sequence in location 17 (the COUNT), what would the machine have answered?

It is easy to see that communication between man and machine is delicate work. It is also important work because of the large role which machines now play in business and science. Preparing correct programs has become a major area for the use of human labor. An enormous demand for skilful programmers has developed, to translate human thoughts into electrical language. To simplify this work, ways have been found to place a greater part of the burden of translation upon the machine itself, by means of other programs permanently memorized by the machine. Such procedures are called automatic programming. They are already remarkably sophisticated, and what the ultimate will be in man-machine relations is impossible to predict. In one very popular system our program of this chapter enters the machine as

```
1     SUM = 0.0
2     DO 3 I = 1, 32
3     SUM = X(I) + SUM
```

Amateur cryptographers will have little difficulty in breaking the code.

PROBLEM 1. Show that if the sequence for location 6 entered the machine incorrectly as 0 1 1 0 1 0 0 0 0, then the sequence in location 2 should ultimately become 0 1 1 0 0 0 0 0 0. This translates to 'copy the sequence now in 0 0 0 0 0 0 into 0 0 0 0 0 0.' The machine would not accept such an instruction. It would stop without punching out a sum, which would be your first indication of trouble. In such a situation it might also have been taught to type out 'faulty instruction in location 2.' What sort of detective work might discover that the fault is really with instruction 6, not with instruction 2 ?

PROBLEM 2. Show that if the sequence for location 5 entered the machine incorrectly as 0 1 0 0 1 0 0 0 0, then the machine would, as expected, punch out a sum and stop, but that the sum would be <u>the wrong sum.</u> This is the most dangerous kind of program error. Unless the machine operator noticed that this program took much less than the expected time to run its course, the incorrect sum might very well be accepted as correct. Suppose an error is suspected. Can you plan a post-mortem? If you ask for the sequence in location 17 (the COUNT) what will the machine answer? What is the incorrect sum which the machine produced?

PROBLEM 3. Adapt our program to the summing of forty-five numbers, X_1 to X_{45}, which have been stored in memory locations 19 to 63. Except for properly storing those forty-five numbers, only one sequence of the program would have to be altered. Which one, and what is the alteration?

Location	New Sequence
__________	__________

10.5 A Reminder.

We've come to the end, and I think that a reminder of what my objectives have been is the most appropriate way to finish. They were:

1. to offer a detailed view of one of the simpler, but important games which make up mathematics;
2. to show that the game is played carefully and honestly;
3. to show that the game is useful; it has applications.

This is typical of the many parts of mathematics, and whatever part you study, you will find it helpful to look for the basic rules (need to know), strategy (nice to know), and applications.

ANSWERS TO PROBLEMS

Chapter 2

Section 2.2

$\bar{Y}$	0 0 1 0 1 0	$X + \bar{X}$	1 1 1 1 1 1
$\bar{X}\,\bar{Y}$	0 0 1 0 0 0	$Y + \bar{Y}$	1 1 1 1 1 1
$\overline{X + Y}$	0 0 1 0 0 0	$XY + \overline{XY}$	1 1 1 1 1 1
$\overline{XY}$	1 0 1 1 1 0	$X\,\bar{X}$	0 0 0 0 0 0
$\bar{X} + \bar{Y}$	1 0 1 1 1 0	$X\,\bar{Y}$	0 0 0 0 0 0
$X + X$	0 1 0 0 1 1	$X + XY$	0 1 0 0 1 1
$Y + Y$	1 1 0 1 0 1	$XY + X\bar{Y}$	0 1 0 0 1 1
$\bar{X} + \bar{X}$	1 0 1 1 0 0	$\overline{\bar{X}Y + \bar{X}\bar{Y}}$	0 1 0 0 1 1
XX	0 1 0 0 1 1	$X\bar{Y} + \bar{X}Y$	1 0 0 1 1 0
YY	1 1 0 1 0 1	$(X + Y)(\bar{X} + \bar{Y})$	1 0 0 1 1 0
$\bar{X}\,\bar{X}$	1 0 1 1 0 0	$\overline{XY + \bar{X}\,\bar{Y}}$	1 0 0 1 1 0

Section 2.3

×	0	1
0	0	0
1	0	1

	∅	P	Q	I
-	I	Q	P	∅

+	∅	P	Q	I
∅	∅	P	Q	I
P	P	P	I	I
Q	Q	I	Q	I
I	I	I	I	I

×	∅	P	Q	I
∅	∅	∅	∅	∅
P	∅	P	∅	P
Q	∅	∅	Q	Q
I	∅	P	Q	I

+	∅	A	B	C	D	E	F	I
∅	∅	A	B	C	D	E	F	I
A	A	A	D	E	D	E	I	I
B	B	D	B	F	D	I	F	I
C	C	E	F	C	I	E	F	I
D	D	D	D	I	D	I	I	I
E	E	E	I	E	I	E	I	I
F	F	I	F	F	I	I	F	I
I	I	I	I	I	I	I	I	I

×	∅	A	B	C	D	E	F	I
∅	∅	∅	∅	∅	∅	∅	∅	∅
A	∅	A	∅	∅	A	A	∅	A
B	∅	∅	B	∅	B	∅	B	B
C	∅	∅	∅	C	∅	C	C	C
D	∅	A	B	∅	D	A	B	D
E	∅	A	∅	C	A	E	C	E
F	∅	∅	B	C	B	C	F	F
I	∅	A	B	C	D	E	F	I

	∅	A	B	C	D	E	F	I
-	I	F	E	D	C	B	A	∅

Chapter 4

Section 4.1 and 4.2

$A + B$	0 1 1 1	$\overline{AB}$	1 1 1 0
AB	0 0 0 1	AB	0 0 0 1
$\overline{A}$	1 1 0 0	$A\overline{B}$	0 0 1 0
$\overline{B}$	1 0 1 0	$\overline{A}B$	0 1 0 0
$\overline{A + B}$	1 0 0 0	$\overline{A}\ \overline{B}$	1 0 0 0

Section 4.3

$B + C$	0 1 1 1 0 1 1 1	$A + BC$	0 0 0 1 1 1 1 1
$A(B + C)$	0 0 0 0 0 1 1 1	$A + B$	0 0 1 1 1 1 1 1
AB	0 0 0 0 0 0 1 1	$A + C$	0 1 0 1 1 1 1 1
AC	0 0 0 0 0 1 0 1	$(A+B)(A+C)$	0 0 0 1 1 1 1 1
$AB + AC$	0 0 0 0 0 1 1 1	$A + (B+C)$	0 1 1 1 1 1 1 1
BC	0 0 0 1 0 0 0 1	$A(BC)$	0 0 0 0 0 0 0 1

Chapter 5

2. $(A + B + C)(D + E) = AD + AE + BD + BE + CD + CE$

4. $(A + B + C)(\overline{A} + \overline{B} + \overline{C}) = A\overline{B} + A\overline{C} + B\overline{A} + B\overline{C} + C\overline{A} + C\overline{B}$

5. $A\overline{B} + B\overline{C} + C\overline{A} = A\overline{B}(C + \overline{C}) + (A + \overline{A})B\overline{C} + \overline{A}(B + \overline{B})C = A\ \overline{B}\ C + A\ \overline{B}\ \overline{C}$
$+ A\ B\ \overline{C} + \overline{A}\ B\ \overline{C} + \overline{A}\ B\ C + \overline{A}\ \overline{B}\ C$, etc.

11.

$A\ B\ C\ D$	$\overline{A}\ B\ C\ D$	$A\ \overline{B}\ \overline{C}\ D$	$\overline{A}\ B\ \overline{C}\ \overline{D}$
$A\ B\ C\ \overline{D}$	$A\ B\ \overline{C}\ \overline{D}$	$\overline{A}\ B\ \overline{C}\ D$	$\overline{A}\ \overline{B}\ C\ \overline{D}$
$A\ B\ \overline{C}\ D$	$A\ \overline{B}\ C\ \overline{D}$	$\overline{A}\ \overline{B}\ C\ D$	$\overline{A}\ \overline{B}\ \overline{C}\ D$
$A\ \overline{B}\ C\ D$	$\overline{A}\ B\ C\ \overline{D}$	$A\ \overline{B}\ \overline{C}\ \overline{D}$	$\overline{A}\ \overline{B}\ \overline{C}\ \overline{D}$

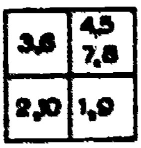
Chapter 6

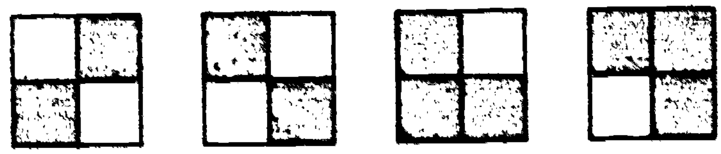
1. My cat has fleas and my dog has fleas; My cat has fleas or my dog has fleas; My cat does not have fleas; AC; $\overline{B}$; $A + C$.
2. V, S, X, Y, Z, T, W, U.
3. Only on Tuesday.
4. On Tuesday, Thursday, and Saturday.
5. I, $\emptyset$, A, A.
6. All do. (See Theorems 21 and 22.)
7. Yes. (See Theorem 34.)

Chapter 7

1. Number 6 belongs to all four subsets; number 9 belongs to none.

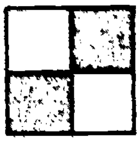
2. 0 1 1 0 0 0 0 0 0 1; 0 0 0 0 0 1 0 0 0 0; 0 1 1 1 0 0 1 0 0 1;
0 0 0 0 0 0 0 0 0 1; 1 0 0 0 0 0 0 0 1 0; 1 1 1 1 1 1 1 1 1 1.

3. 

4. 

5. Theorem 12.

6. Theorem 18.

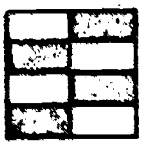
7. 

Theorem 21

8. 

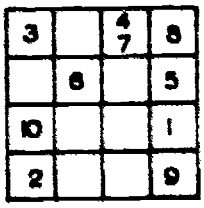
Theorem 38 (see Problem 9).

9. Theorem 38 (see Problem 8, also).

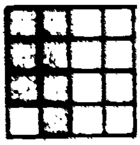
10. 

Theorem 39.

11. (c).

12. 

1	$\bar{A}\ \bar{B}\ \bar{C}\ \bar{D}$	6	A B C D
2	$A\ \bar{B}\ \bar{C}\ \bar{D}$	7	A B C D
3	$A\ B\ \bar{C}\ \bar{D}$	8	$\bar{A}\ B\ \bar{C}\ \bar{D}$
4	$\bar{A}\ B\ \bar{C}\ D$	9	$\bar{A}\ \bar{B}\ \bar{C}\ \bar{D}$
5	$\bar{A}\ B\ C\ \bar{D}$	10	$A\ \bar{B}\ C\ \bar{D}$

13. 

14. Bend the cylinder into a doughnut shape.

Section 7.5

7 on second try, 11 on third; 9 successes after a success; 2 failures after a success.

15. Opium eaters don't wear white kid gloves.

Chapter 8

1. Left to right, -, +, x; 2. Theorem 10, Theorem 8.

3. $\bar{A}$; 4. $AB + \bar{B} = A + \overline{\bar{A} + B}$; 5. A + B; 6. $\overline{A + B}$

7. $A + \bar{B}$; 8. AB + AC + BC; 9. $(A + B + C)\overline{ABC}$.

Chapter 9

1. 1 1 0, 1 1 1, 1 0 0 1, 1 0 1 0, 1 1 1 1, 1 0 0 0 1, 1 1 1 0 0, 1 1 1 0 1 1.

2. 11, 12, 20, 25, 31, 36, 45, 63.

3.

3	1
2	0

6	2
7	3
5	1
4	0

12	13	5	4
14	15	7	6
10	11	3	2
8	9	1	0

4. 1 0 1 0 1, 1 0 1 1, 1 1 0, 1 0 1 1 0 0.

5. 1 1 1 1 0.

6. Record: 0 1 1 0 1 0 0 1

   Carry: 0 0 0 1 0 1 1 1

   This full-adder uses more equipment than the one in Section 9.4.

7. Record $= ABC + (A + B + C)\overline{AB + AC + BC}$

   Carry $= AB + AC + BC$

   This full-adder uses less equipment.

Chapter 10

1. Check the COUNT. It will be zero.

2. COUNT = SUM = 207.

3. Location 15: 0 0 0 1 0 1 1 0 1

REFERENCES

1. Geo. Boole, "The Mathematical Analysis of Logic", Cambridge, 1847 (reprinted by Blackwell Publ.)

2. Geo. Boole, "An Investigation of the Laws of Thought", London, 1854 (reprinted by Dover Publ.)

3. F. Scheid, "Elements of Finite Mathematics", Addison-Wesley Publ., 1962

4. H.G. Fless, "Boolean Algebra and Its Application", Wiley Publ., 1964.